DOCUMENT RESUME

ED 148 297 IR 005 207

TITLE New Directions in Educational Computing. Proceedings/1977 Winter Conference.
INSTITUTION Association for the Development of Computer-based Instructional Systems.
PUB DATE Feb 77
NOTE 324p.; Proceedings of the 1977 Winter Conference of the Association for the Development of Computer-based Instructional Systems (Wilmington, Delaware, February 21-24, 1977)

EDRS PRICE MF-$0.83 Plus Postage. HC Not Available from EDRS.
DESCRIPTORS *Computer Assisted Instruction; *Computer Oriented Programs; Course Evaluation; Deaf Education; Elementary Secondary Education; *Futures (of Society); Health Occupations Education; Junior Colleges; Music Education; Problem Solving; Shared Services
IDENTIFIERS PLATO

ABSTRACT

This collection of 39 conference papers is arranged by subject areas: keynote address; general topics; computer-based musical instruction; elementary, secondary, and junior college computer uses; health sciences computer uses; futures for courseware sharing; PLATO projects and uses; computer implementation issues; computers and the deaf; and evaluation by computer. (DAG)

Documents acquired by ERIC include many informal unpublished materials not available from other sources. ERIC makes every effort to obtain the best copy available. Nevertheless, items of marginal reproducibility are often encountered and this affects the quality of the microfiche and hardcopy reproductions ERIC makes available via the ERIC Document Reproduction Service (EDRS). EDRS is not responsible for the quality of the original document. Reproductions supplied by EDRS are the best that can be made from the original.

ED148297

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

Proceedings/1977 Winter Conference
Wilmington, Delaware

New Directions In Educational Computing

ADCIS

SIGILLUM UNIVERSITATIS DELAWARIENSIS 1833

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY
Fred T. Hofstetter

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC) AND USERS OF THE ERIC SYSTEM."

IR005 207

February 21-24, 1977

TABLE OF CONTENTS

Page

I. KEYNOTE ADDRESS: The Inaugural Dean Lecture

"Why CBI? An Examination of the Case for Computer-Based Instruction," Peter M. Dean 1

II. GENERAL TOPICS

"Computer Assistance in the Social Processes of Learning," Charles A. Weaver and Bonnie Anderson Seiler 26
"CASPER: An Interactive Approach to Computer-Managed Testing," Charles Guerra, Harlan McDaniel, Joseph Kaufman, and David Fuller 39
"CMI and the Logical Analysis of Curriculum," Robert F. Yeager 55
"A Minicomputer CAI System," Thomas T. Chen, Douglas W. Jones, A. B. Baskin, Larry Sherman, and Allan H. Levy 61
"1/2 of 1/2 - Greater Than the Whole?" Leland A. Szklennik 75

III. NATIONAL CONSORTIUM FOR COMPUTER-BASED MUSICAL INSTRUCTION

"Two Important Principles for Constructing Computer-Based Music Instruction Programs--The Circle of Fifths, and Key Transposition," Marvin S. Thostenson 80
"The Complete Computer-Based Music System: A Teaching System - A Musician's Tool," G. David Peters 93
"Guidelines for the Development of Computer-Assisted Instruction Materials in Music Theory," Michael Arenson 101
"Interactive Simulation/Games as an Aid to Musical Learning," Fred T. Hofstetter 104

IV. ELEMENTARY/SECONDARY/JUNIOR COLLEGE INTEREST GROUP

"A Computer-Managed Information and Instructional System for Vocational and Technical Education," Nevin R. Frantz, Jr., and Edward E. Boas, Jr. 118

TABLE OF CONTENTS (continued)

"An Instructional System in Mathematics," Catherine E. Morgan 129
"The Teacher and Technology-The Woodridge Study of CAI-A Three Year Study," Katherine Hirschbuhl 133

V. HEALTH SCIENCES INTEREST GROUP

"Faculty and Module Evaluation Via PLATO," Arthur D. King, Stephen J. Summers, and Marie E. Shanley 144
"Resident Physician Performance on a Patient Management Computer Simulation," James V. Schultz, Robert S. Newsom, Steven Entine, Judith Neal, and Richard B. Friedman . . . 154

VI. POSSIBLE FUTURES FOR COURSEWARE SHARING

"CAI System Transparency -- One Solution," James A. Schuyler 160
"Courseware Transport -- When and How?" R. A. Avner 162
"Problems in the Future of Courseware Sharing," Stephen K. Lower 164
"Transportability and Sharing -- The Need for Standards," J. W. Brahan 167
"The Problem of Natural Language Barriers," Bert Camstra 170
"A Case Study Using MUMPS Examples [Condensation]," Richard F. Walters 173
"Possible Futures for Courseware Sharing - A Role for Institutions," Robert G. Votaw 176
"Possible Future for Courseware Sharing," Thomas Chen, and Vincent H. Wu 181
"Sharing Courseware," Alfred Bork 183

VII. PLATO INTEREST GROUP

"A Glimpse of the PLATO V Terminal," Jack Stifle 186
"Factors Contributing to Success in Author Training," Larry Francis 192
"Internal Evaluation of the PLATO System," R. A. Avner 197

TABLE OF CONTENTS (continued)

"The Control Data Learning Center Network," Pauline R. Jordon 200
"Overview of the PLATO Modern Hebrew Project," John D. Eisenberg 202

VIII. IMPLEMENTATION INTEREST GROUP

"Optimizing Coursewriter Parameters To Reduce Response Time at Remote Sites," R. Lindwall 222
"A Community of Individuals: Cooperation and Individualization in Computer Science Education," Kenneth Modesitt 234
"The Computer Display as a Medium in the Teaching of Aesthetics in Visual Design," Raymond D. Nichols, and James H. Wilson . . 248
"Comprehensive Computer Based Education Development in a Network Environment," James E. Eisele and John S. Edwards 256

IX. INTEREST GROUP OF EDUCATORS FOR THE DEAF

"A National Survey of the Use of Computers for Instruction in Schools for the Deaf," James R. vonFeldt 268
"Computer-Assisted Instruction at Minnesota School for the Deaf," Benet R. Freund . . 279

X. EVALUATION

"A Computer-Implemented Modeling and Problem-Solving Approach to Teaching Statistics," Stanley W. Steinkamp 281
"Learner Control and CAI: Assumptions, Findings, and Questions," Esther R. Steinberg 292
"Data Analysis and Retention: Experience and Recommendations," Paul Lorton, Jr., and Rosemary N. Killam 298
"Bootstrapping CAI," Richard G. Montanelli, Jr. 308

WHY CBI?

AN EXAMINATION OF THE CASE FOR COMPUTER-BASED INSTRUCTION

Peter M. Dean

I am deeply honored by the action of your Steering Committee naming this keynote lecture for me and inviting me to deliver this first iteration thereof. This honor, by implication, suggests the role of an old timer in ADCIS for myself. I assure you, however gray tho I may be, an oldster I'm not.

It is true that I have attended a number of ADCIS, ADIS, and 1500 Users Group Meetings over the years. As a matter of fact, I don't believe I've missed one. During that time, I have had hundreds of conversations about computer-based instructions with many of you. I have viewed innumerable demonstrations and I have filed reams of paper and handouts.

I've also been fortunate over this priod that my employer has permitted me to work with computers and computer-based instruction for some 10 continuous years. First, with a 1500 System that I installed myself and, more recently, with Coursewriter III and the Interactive Training System. And, for the past year, with the new Interactive Instructional System. As a result of these activities, I've had a great deal of fun, met some fascinating people and, I believe, learned a few things about Computer-Based Instruction.

One of the things that I have learned is that computers are fun, they're seductive, they're demanding, they're rewarding. You feel good when they do what you make them do. It is this very characteristic that has caused us many of our problems. We have often become so very enamored with our machines and software that somehow we never quite got around to doing what the external world would term "useful work."

This then is the theme of this lecture. I'm going to share with you my perception of what is important and worthwhile about computer-based instruction. I'm going to tell you why I feel as I do and I am going to cite a little research from time to time, but mostly, I'm going to make assertions based upon personal experience and reflect a very pragmatic view of the world.

"Why CBI?" I will expand that question. Given 3,000 years of education experience dating back to the Greeks, given the wonders of educational television and graphic design, given the educational perceptions of Norman Crowder and B. F. Skinner and their followers, why on earth do we few keep diddling aroung with logical machines and pretending that we have something better? The answer to me is clear ... we do.

To explain what it is, I'll have to go back some 16 years to Santa Barbara where Norman Crowder had developed the Autotutor® and the Tutortext®. The Tutortext is perhaps better known to you as the scrambled book. One day while working my way through some difficult material on alternating current circuits in a Tutortext manuscript I discovered that I had for some period of time been reading and turning pages, but had ceased to learn. I stopped and thought about what had happened.

As I reconstructed the experience, the sequence of events was somewhat as follows:

> In the beginning, when I had encountered the question at the bottom of a page of explanation, I had dutifully attempted to select the correct answer and if I wasn't sure, I had re-read the explanatory material or paragraphs and made sure that I had made the best selection I was able.
>
> Then, when I turned to the answer page and discovered that I was wrong, I dutifully read the remedial material and went back to the orginal question and again put my full effort in trying to understand the question and deriving an answer on my own.
>
> As time passed, I ceased to review quite as carefully and began to guess on answers. And a little later, I began to read the question before I even read the explanatory material to see whether I could just answer the question quickly or at least have some idea as to what the correct answer might be.

If I selected a wrong answer, I didn't bother to read the remedial material at all, but immediately returned to the question page and selected the next most likely alternative. I rationalized such behavior for myself on the basis that I "really" knew the material and had selected the wrong alternative because I was kind of slopy in reading the question, rather than for any lack of understanding. What was missing in my experience with the Tutortext and later with Skinnerian programmed instruction?

My initial explanation to myself was that I was lazy and undisciplined. Unlike other people, I was unwilling to commit myself to doing what had to be done if I was to learn. I tended to regard myself as an exception and continued to feel that program instruction was a suitable vehicle for the instruction for other more perfect individuals.

But, I had my doubts, and I began to watch others as they used programmed materials. Indeed it seems other people are very much like me. When it comes to situations that involve what is preceived as being "hard work", we humans tend to be labor avoiding mechanisms.

> This seems to be what happens to students when they are working their way through organized instructional material. They begin the experience with the greatest of good will and intentions, but depending on the individual, anywhere from 5 minutes to 50 hours after they begin, they begin to take short cuts, they look at questions and then the answers, and then say to themselves, "oh, yes, of course, that's the answer I would have choosen."

Educational theory suggests that learning does not take place unless the learner himself is overtly involved in some interaction with that which he is attempting to learn. An interaction which only involves recognizing or assuming that one knows an answer does not result in that answer being learned.

The missing ingredient in programmed self-study, and I use the term "program" here in its broadest sense, is external discipline. As long as the learner himself is the only person immediately responsible for his learning activity as he moves from idea to idea, from concept to concept, from problem to problem, I can assure you that on a continuing basis very little learning will occur.

I will cite another analogy.

Consider John Hopkins at one end of a log and a student at the other. We're talking about the tutorial interaction now. What is it that makes this an "ideal" learning situation? Is it the fact that the tutor can listen to the student's response, very carefully analyze what the student says and provide him with selected additional bit of information or leading question and, hence, lead him to make his own discoveries? I don't think so.

My experience says that the important thing in the tutorial interaction is that somebody external to the learner is insisting that he wrestle with the subject matter and that he answer the question. It is not the very cunningly contrived question which causes learning to occur.

I'll cite you some evidence.

It has been demonstrated frequently that students can tutor one another quite effectively as far as learning is concerned. Certainly, a student tutor is no master teacher, no Socrates in disguise, he is just another body. Yet, the technique works. Why does it work? Because of external discipline. I repeat my original question now ... Why Computer-Based Instruction? The answer ... because it can provide external discipline.

Am I being simplistic in my analysis of the values of computers in instruction? Before I address myself to the question of how does one best go about using computers to apply external discipline, let's look at some other experiences. Experiences which should be well known to you and indeed part of your every day life. Let me ask a retorical question. When does a college student study? The answer ... a college student studies immediately before examinations and very diligently, indeed, immediately before final examinations.

I suspect that if one where to take an "opinion survey" amongst college students and ask them to describe an optimum strategy for learning in the college environment, that one would not find very many students suggesting that the procedure which is followed is the one which should be followed.

I suspect that they, like those of us who are advising people who are students, would say ... "No, you study consistently throughout the term; that you should learn each item, each concept thoroughly as you go and then when you come to finals time you are going to have it all in your head then finals will be a snap." Yet, that's not the way human beings behave. Human beings on the whole, wait until final week and cram like crazy.

If one could cause a student to work as hard during the remainder of a term as he works during that period that is immediately associated with finals, one would almost be tempted to say that four years of college could be completed in a year's elapsed time.

I suspect that does not actually follow because the amount of work that an individual can do is, I believe, to be rather severly limited and if one were to work very diligently for an hour a day that would be about the limit of a human's capacity. I don't know what the number is, but I'm certainly sure that it is not 8 hours a day for 13 weeks - the length of a college semester.

Those of us who have had experience with elementary and pre-collegiate education know that we do not use a final examination approach to cause students to study and to work, instead we do our testing somewhat more frequently and somewhat less threateningly. Nevertheless, the human behavior remains the same from the time that we first find children in formal education through situations such as my own, where we are training people in middle life technical skills so that they can continue career development and growth.

Let me turn now from everyday experience and everyday knowledge to research. The Summer 1976 issue of the REVIEW OF EDUCATION RESEARCH published by the American Education Research Association contains a fascinating review entitled, "Behavorial Instruction in the College Classroom" by Professor Arthur L. Robin of the University of Maryland at Baltimore County. Professor Robin reviewed some 39 different studies of the use of the Keller Plan or behavorial instruction at the collegiate level.

"Five features characterize behavorial instruction in its original form:

1. The go-at-your-own-pace feature which permits a student to move through the course at a speed commensurate with his ability and other demands upon his time.

2. The unit-perfection requirement for advances which lets the student go ahead to new material only after demonstrating mastery of that which preceded.

3. The use of lectures and demonstrations as vehicles of motivation rather than sources of critical information.

4. The related stress upon the written word in teacher-student communication, and

5. The use of proctors which permits repeated testing, immediate scoring, some almost unavoidable tutoring, and a marked enhancement of the personal-social aspects of education process.

I continue to quote from the article ...

> The instructor incorporates these features into a course by breaking the reading material down into a series of small units, in preparing materials for each unit which consists of instructional objectives, study guide questions, and clarification of ambigious points. A student reads the unit materials and learns the answers to the study guide questions. He then comes to class where he takes a written quizz consisting of a subset of the study questions themselves or closely-related questions. The proctor evaluates his answers and provides immediate feedback. If the student has answered the questions to a high-level mastery, typically above 80%, he can advance to the next unit at his own pace.
>
> If he fails to reach the mastery criterion, he re-studies materials and re-takes the quizz until he does reach criterion. Grades are based primarily upon the number of units mastered with a small percentage based on mid-term

and/or final examinations. The proctors are typically advanced undergraduates who have recently completed the course at high-level mastery." Since the original definition of the Keller Plan, a number of variations have appeared. Nevertheless, the concept of short study units, frequent testing, and procedure only after demonstrating mastery continue to be the key characteristics of behavorial instruction.

Robins' review indicates consistent improvement in performance, improved retention, and positive attitudes on the part of the students in the studies. Indeed, the only negative finding, if indeed it is negative, is that the dropout rate or withdrawal rate tends to be higher in behavorial instruction classes than those in which the student need only attend lecture and take examinations.

The length of time the student is actually engaged in studying tends to be constant. Comparing self-study time, preparation for examinations and attendance at lecture time for conventional class with self-study time and quizz taking time for the Keller Plan class indicates that the total time elapsed is the same.

Note the key characteristic of the Keller Plan. The student studies a little bit, not terribly long, just a unit, and he takes a test. External discipline is applied.

Note also that a simplistic model of instruction is used. The instructor breaks pre-existing reading material down into small units and writes explanatory materials for those concepts which are obtuse. No great effort is put into attempting to crawl into the students mind and understand his perception of the world in order to prepare instructional materials which are uniquely tailored to his needs.

Robin or at any rate those involved in behavorial instruction make rather a point of the fact that a human being is used as the proctor rather than some dispassionate machine. I suggest that this may be an overstatement or may be reflecting a misunderstanding of the capacity of a well-programmed CBI interaction.

Let me turn now to some evidence internal to the CBI fraternity if you will. My colleague, Harvey Long, has been insisting for several years now that "drill and practice" is where it's at. I concur in part. "Drill and practice," indeed, is one of the places where it is.

Consider if you will the most basic of the drill and practice applications - the flash card emulator. Manual flash card sessions consist of the learner sitting across the table from the presenter, who holds in his hands a series of cards testing the learner's knowledge of arithmetic facts, or spelling, or language translation words, or what have you.

Consider the simplest cases, a person working with an elementary school child learning multiplication facts. The presenter offers a problem to the student, the student answers either correctly or incorrectly and the presenter puts the problem card in either the good stack or the bad stack depending upon the answer.

Those items in the bad stack are recycled again and again until such time as a student has worked his way through the possible combinations for which he is held responsible.

An old learning technique one which is quite effective, one which requires the expense of one individual human per each student. The human offers some positive aspects in that if a student is having trouble, he can selectively sort problems so that the student gets items that he has mastered as well as those which he has failed yet to master. On the other hand, the proctor also becomes empathic with the student and begins to by voice, mannerism, indeed, by actual conduct attempt to help the student who is having difficulty and I am afraid in many instances turns out to be not the student's friend, but the student's enemy in that he so clues the student that learning fails to take place. The proctor should be demanding demonstrable mastery.

The individual presenter being a human also tends to reflect through body attitude, facial expression, tone of voice, displeasure with a student who is having learning problems as well as pleasure with a student who is doing very well. Unfortunately, it is those students who are having difficulty and who most need positive reinforcement who are going to receive a negative social feedback from the presenter.

Computer-based drill and practice, on the other hand, offers a completely dispassionate presenter. A presenter who does not become impatient or unhappy when the student seems to have particular difficulty remembering that 7x3 is 21. One is certainly not surprised then, that experience with students who have had learning difficulties in the past, show that computer-based drill and practice is a useful, indeed, a very powerful tool.

Note that the drill and practice algorithm insists that it is a two-party transaction. You don't hand a set of flash cards to a student and expect him to work his own way through and successfully learn his timetables. Even college students memorizing formulas in organic chemistry find it more useful to work with a colleague taking turn and turn about, rather than sitting in a corner attempting to administer flash cards to themselves. It is just too easy to peek. Again, my basic premise illustrated: external discipline is necessary if learning is to occur.

I'll step aside now from the main thrust of my argument for a small excursion in the field of mythology. Ten years ago when I first began to work in the field of what was then called computer-assisted instruction, there was a premise floating around concerning a direction that we were all striving to achieve. That direction was a model of a world which ... well, let me describe it to you ... in that world we perceived a student coming to school and setting at a terminal ... and it would be a very rich terminal indeed ... with computer-provided audio and with lots of graphic capacity. We suspected that in the future we would have full color graphics, it was going to be a very handsome, rich kind of a terminal. And as the student sat down and began to study and learn, the program itself would also be learning. Learning about how the student learns, what the student's learning style was about, how best to present material to the student so that he would be able to most effectively use it.

It seems that those of us, and I must number myself among them, who had that image of the future were prone to assume a touch of the deity. Somehow we were going to be something greater than human, we were going to be able to crawl inside another person's head and decide what was right and wrong for him because we would have such a great understanding of how his mind worked.

Well, when I introduced this paragraph, I suggested that it was an excursion into mythology and indeed this proved to be the case.

One more side discussion before I return to the gist of my argument. This one might best be titled, "The Cost of Doing Business." Again, returning to the good old times, one of the problems was that it was costing us anywhere from $2.50 to $12 per student-terminal hour to provide CAI services. We were assured that this was going to change in the future.

Don Bitzer at the American Association of Advanced Science Meeting in Denver in 1960 assured us that something on the order of 25 cents per student hour was going to be achieved in a very short time. And Don was not the only person who was optimistic about the reduction in expense in CAI.

And indeed the cost of providing computing facility has decreased as time goes on. Unfortunately, a corresponding reduction in the cost of providing data communications has not occurred. If anything, telephone rates have continued to inch their ways upward in parallel with general inflation. Even the advent of standalone minis such at the IBM 5100 has not provided an answer to the problem of the expense of providing facilities for students.

Consider also that manufacturing expense represents only a small part of the total expense in taking a product from the engineering laboratory to the ultimate consumer. As a number of small firms have discovered to their distress, marketing, distribution, and field support are very expensive services to provide. Yet, are absolutely essential for success in the marketplace. I would suggest that current experience indicates that a decent television set is obtainable at retail at something between $300 and $400 and that the total market for color television sets is somewhat greater than the potential market for CAI terminals. I should be very surprised if within your lifetime or mine or indeed at any time, that the equivalent expense of a single user terminal is going to be much less than a $1,000. Such a price implies a very large capital outlay, should we attempt to provide for every learner a terminal for his use whenever his learning might be augmented by the interaction with the device. I am led to the conclusion that access to computer terminal will always result in some contention, i.e., every student will not be able to go to a terminal whenever it is convenient for him to go to the terminal.

Furthermore, I believe that people who plan education or training systems will have to plan that the student require access to a terminal for only some portion of his total experience rather than continuously.

I'll summarize points of my argument so far.

1. Humans, all of us, are more apt to learn if some external discipline is applied to the learning process.

2. External discipline can be applied by 2 operators. Other humans or their surrogate, the logical machine - the computer.

3. A computer is a better surrogate for an instructional designer, or professor, than another human who is untrained and who has his own values and brings to the instructional interaction emotion or lack thereof in varying unmanageable degrees.

4. The computer resource is a rare resource. It is one which should be used judiciously so that its benefits can be applied to as large a number of students or learners as possible.

These observations lead me to a challenge. Question: what is the optimum strategy for the use of computers in the education or training environment? I will describe my strategy and then illustrate it with some work which has been done internally within IBM.

As is the case with most models which must be widely replicated, it is simple and designed to be easily understood by a variety of practitioners. One does not have to be an educational technologist to understand how to use this tool.

The learner is asked to study on his own for somewhere between 2 and 4 hours. He is then directed to a computer or terminal where a unique examination is administered. The number of questions usually ranges between 15 and 30. A passing score is 90% or better or in many instances 95% or better. On-line remediation and/or on-line remedial assignments are provided tailored to the specific items that the student failed. If the student passes the examination he is directed to move on to the next unit of study, again conducted off-line. If he fails, after he has done some remedial work, he is asked to come back and attempt examination again. Since the examination is unique each time it is administered, knowing the answers to the questions encountered the first time does not assist the student in passing the second. In this model the student may attempt the examination twice. In order to try it a third time, he must go to an instructor or the person responsible for the course and receive permission and justify his lack of progress as demonstrated by his inability to pass the gating examination.

A few words about the construction of such an examination. At the beginning of the off-line self-study experience the student is told specifically what skills, knowledge or abilities he will have mastered at the end of that unit of study. In other words, we state in behavorial terms the learning objectives of the unit. Each question in the examination is associated specifically with a learning objective. In our implementation, the objective being tested appears at the top of the screen before the question itself appears. Each question is specifically related to a learning objective. The same objective may be tested by one or more questions. Typically, each time an objective is tested, there are 5 alternative forms or alternative questions which test that objective. Thus, if an objective is tested 3 times in the examination, there will be 15 different questions which may be used to test that objective. When the individual student is at his terminal to take an examination, one of the five forms for each question is randomly selected to be administered at that session. The likelihood of a student obtaining the same sequence of questions over a 20- to 30-item examination is vanishingly small. The examinations are secure. Even if a student were to collect copies of the examination from other students assuming a hardcopy printer, or if he had access to the master file, the total number of questions is so large that if a student were to memorize the answers to all of them he would have done exactly what we want him to do. He would have mastered the objectives quite thoroughly. If the student would rather study questions than study learning materials structured to assist in learning ...well, that's his choice.

Although we have used several different models in administering these examinations, my favorite, is one in which the student on being presented with an objective receives a question. If he passes the question, he then moves on to the next objective. If he fails the question, he receives another question from set designed for the first objective. In order to have failed an objective, it is necessary for him to miss two questions rather than one. This model permits us to insist upon an objective passing criterion of 95%. That is, the student must be nearly perfect in mastery of objectives before he is permitted to go on to another unit. Theoretically, I feel that if a student should not be nearly perfect, he should be absolutely perfect. However, recognizing the difficulty of writing goood questions, I am quite satisfied to permit him to miss one or two objectives in a large examination, and one objective in a short examination.

A comment about the off-line self-study materials. It may range from the very prosaic, i.e., "go read chapter 15 in your textbook" to some rather esoteric things. Certainly, you can use programmed instruction for short periods of time. You can use audio tapes, you can prepare video tapes, you can ask people to go to television monitors and look at video tapes or look at motion picture films, you can ask people to go to self-directed laboratories using the audio tutorial approach. There is really no limintation on what you can do with a self-study session.

Indeed there is no prohibition about against using the computer terminal itself to provide simulation or drill and practice exercises or other applications for which the machine is an appropriate vehicle. However, the key use of the machine in management of the instructional process is the periodic gating and the provision of an appropriate remediation on the basis of specific items a student failed to comprehend or master.

The kind of on-line examination that I have described is unquestionably the easiest kind to develop, to compile, and to administer. However, there are other ways of administering on-line examinations using other examining vehicles. And without going in to detail, I would suggest that, for example, in a Cobol Language Programming course that an examination might very well consist of a requirement that a student find the bug in a program or later on, as his knowledge of the subject became more extensive, code a subprogram or code a piece of a program. Toward the end of the course where one is concerned with putting together all of the skills which are learned, the examination might be a requirement that the student write a program which would produce specified output. The student's success would be measured by an examination of the output. Similarly, in other situations, I can image that successful passage of an examination would be demonstrated by the ability of the student to interact with the simulation and manipulate the variables such that the desired output was obtained.

Don't look narrowly at the question of examinations. Be more concerned that the experience, the skills that you are requiring the student to exhibit when you test him are measuring the objectives that you intended that he should have achieved when he was learning. Several questions can be asked about the approach I have described. I would like to consider them.

One question is "Have I not just described what everybody understands as computer-managed instruction?" My answer, "No, I have not." Computer-managed instruction implies that a person will by observing the student's performance and tailor instruction to the student's learning needs. I have not described nearly so complex an application. The learning materials in my model are essentially linear in nature. You start at the beginning and progress as rapidly as you can until you come to the end. Incidently, in my model there is no prohibition against a student who believes he is competent to look at the learning objectives, go immediately to the terminal and take the final examination and move on until he finds his appropriate place in the course.

Second question is .. Have I not described a very punitive instructional system? My answer is "No, I don't believe so. Certainly, it is not used that way." The function of the examination is not to penalize a student or tell him that he has failed or in anyway be negative about him. The function of the examination is to enable the student himself to know that he has mastered the material that he was expected to learn. And, the way we use it, we put pressure on the student to demonstrate how rapidly he can get to examinations and how quickly he can learn his material.

We suggest to the student that what we want him to do is to demonstrate his ability to master certain material. The reason that we do not permit a student to attempt an examination more than twice without external intervention by another human is that we do not wish students to be trapped in learning segments where they are lost. One of the problems of self-study is that a very diligent student can get himself in a box some place and you don't find out about it for too long a period of time.

Another question which is asked is, "Are you not misusing the power of the computer?" Here's this very powerful logical machine capable of doing all kinds of wonderful things and you are using it in a very simplistic kind of fashion. My answer is that I would certainly much rather use the machine in this way than a human being. And, in this particular kind of application, we're really playing to the machine's strength. It is a superlative recordkeeper and incredibly patient. It is perfectly willing to administer examinations from now until doomsday and never complain.

One other observation before I close. Given a three-hour off-line to approximately half-hour on-line ratio, one terminal can be counted upon to optimally serve seven learners. Give the further observation that people do not study continuously, one would suggest that perhaps three hours of off-line study might occur over an elapsed time of 4, 5 or 6 clock hours. The ratio of machines to learners is further improved as are the economics of the case. We have an internal rule-of-thumb which says a location where students are using terminals receives an additional terminal whenever 70% of the available terminal hours are used. This seems to be quite handy and has not resulted in serious queing problems.

My answer then to the question, "Why CAI or CBI?" is: It is a valid answer to educational problems. With the use of computers, students can learn more effectively and more productively and perhaps, even more importantly, use the one tool which can assure that every learner no matter how bright or how dull learns that which he is set to learn.

Thank you very much.

COMPUTER ASSISTANCE IN THE SOCIAL PROCESSES OF LEARNING

Charles A. Weaver
Computer-Based Education Research Laboratory
University of Illinois
Urbana-Champaign, Illinois 61801

and

Bonnie Anderson Seiler
Delaware PLATO Project
University of Delaware
Newark, Delaware 19711

ABSTRACT

Computer assisted instruction has traditionally been associated with individualized instruction. While there is a great need for such instruction there are also dangers associated with it.

Many observers have noted that it is important for students to verbalize what they have learned and to interact directly with teachers and fellow students about materials being studied. Great damage can result when individualized instruction is carried out in a situation in which social interaction is lacking.

Computers can be used effectively to aid the student communication process as well as to individualize instruction. In this paper we discuss various ways in which computers can facilitate student-student interaction and also can direct students to deal with one another's written ideas.

Examples include lessons in which students learn from each other's strategies and moves, work cooperatively to solve a common problem, pose problems for other students to solve, display their work for others to use, and exchange questions and comments about subject matter with teachers. Specific examples are taken from materials developed by the authors as part of the PLATO Elementary Mathematics Project.

INTRODUCTION

More and more, computers are being used to provide instruction tailored to student needs. The learner sits alone at a terminal, working at his own pace. He receives immediate feedback from the computer as it modifies the material according to his errors. This attention is much more constant and individualized than that which a classroom teacher can provide. Such applications of computers hold great promise for the future of education.

At the same time there is the specter of thousands of students sitting in little cubicles, busily interacting with computer terminals - seldom with other human beings. It is clear that such a situation is undesirable, even from the point of view of efficient learning of actual material. It is important for students to verbalize what they have learned and to interact directly with teachers and fellow students about materials being studied. Concepts need to be clarified by discussions. Problems need to be worked on with teachers and peers in order for problem solving techniques and suble understandings of material to be exchanged.

Teachers and students are very much aware of this need to communicate and share ideas in order to learn. Accordingly, many classroom activities are devised to serve this purpose: question-and-answer periods, class discussions, group projects, notebook keeping, and displays of student work. Students work together to solve problems and to help each other understand what they are learning.

Effective use of student communication as an instructional tool is, in many ways, just as difficult and time consuming to the teacher as is the use of individualized instruction. The teacher must supervise the activities, guide discussions, correct obvious errors, keep students on the right track, and see that the quiet student is not excluded. Having students look at each other's work involves much keeping track of papers, posting of lists, and managing of books. Often the learning becomes lost or ignored in the process.

Computers can be used effectively to aid the student communication process as well as to individualize instruction. They can make it easy and attractive for students to work together on problems. Computers can aid in the bookkeeping and display problems of exchanging student work. They can ensure that each student is involved. They can make it easier for students and teachers to communicate with one another.

In this paper, we describe some of the ways in which we have been able to use computers for student communication as part of the PLATO Elementary Mathematics Project at the University of Illinois, Urbana-Champaign (Seiler and Weaver, 1976)

We will describe first, several computer activities which serve to involve the student directly with one another; second, lessons in which students deal with the ideas and contributions of other students who are not using the computer at the same time; and third, ways in which student-teacher interaction can be assisted by computer.

DIRECT INTERACTION - STUDENTS USE THE COMPUTER TOGETHER

Computer games played by two or more people provide opportunity and motivation for students to use mathematical skills. Well constructed games require the student to pay attention to his opponent's moves and motivate him to learn from them. Computer games can make use of non-trivial mathematics, because the computer can act as referee and also see that the rules are followed (that is, whether the mathematics is being used correctly). With a computer there is no need to appeal to the teacher for a decision.

Other advantages of computer games include their constant availability; more than one game can be going on at one time and none of the component playing pieces is missing. The computer can easily provide a variety of options to the players, such as whether to follow the simple or the advanced rules, and adjust the game board accordingly.

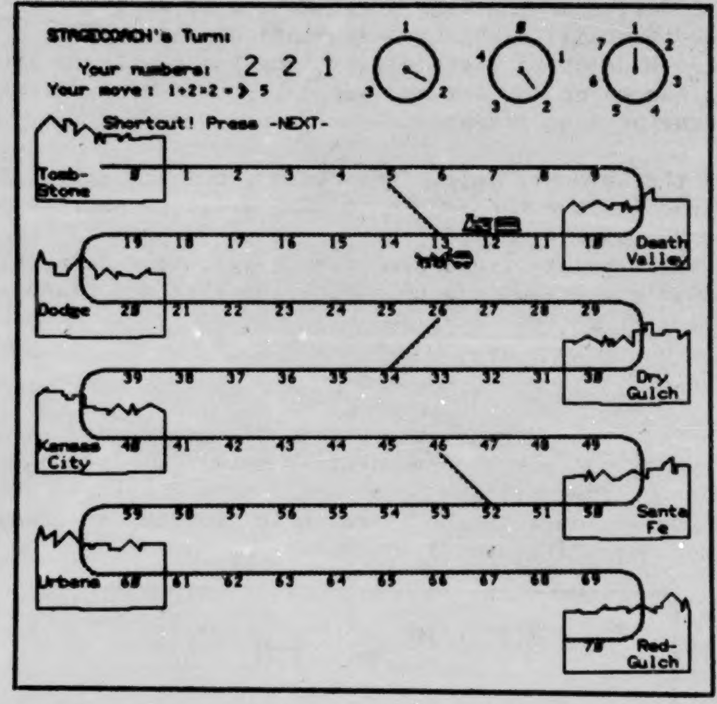
Most of our games were designed for two students to play together at the same terminal. In this way they can discuss possible moves and learn from one another's methods and strategies. If a second player is not available, the student may play against the computer. When the computer is the opponent, the game loses a great deal of spontaneity but usually gains a richness of playing strategy. There is also the capability of permitting a student at one terminal to play against someone at another terminal, perhaps located hundreds of miles away. While this greatly expands the pool of available players with a variety of strategies, the remoteness does not allow the opponents to talk with each other about moves.

Any game, such as a student vs. computer game, which has not been written to foster direct student-student interaction can be adapted simply by having two students work together as a team against the computer. In the case of an interterminal game, two students at one terminal can team against two students on another terminal. Players then benefit from direct interaction and a variety of opponents.

We describe two games we have used.

(1) How the West was One + Three x Four.

"..West.." is a board game which provides experience in generating and evaluating arithmetic expressions and general gaming strategy (Figure 1). In the game a stagecoach and train race along a numbered track. Players alternate moves. Each player determines the size of his move by first spinning to obtain three numbers and then using the three numbers to generate an arithmetic expression. The length of the move is the value of the arithmetic expression. In order to make the move the student must compute the value of his expression correctly. The expressions are subject to restrictions: each of the 3 numbers must appear exactly once and no arithmetic operation (+,-,x,÷) may be used more than once. For example, if a player spins and gets 2,3, and 5 he can make 3x2+5=11, or 5x(2+3)=25 but not 5x3x2=30. The strategy of the game is complicated by a number of special move conventions, such as shortcuts and bonuses for landing on certain squares.



STAGECOACH's Turn:

Your numbers: 2 2 1

Your move: 1+2*2 => 5

Shortcut! Press -NEXT-

Tomb-Stone
Death Valley
Dodge
Dry Gulch
Kansas City
Santa Fe
Urbana
Red-Gulch
0 1 2 3 4 5 6 7 8 9
19 18 17 16 15 14 13 12 11 10
20 21 22 23 24 25 26 27 28 29
39 38 37 36 35 34 33 32 31 30
40 41 42 43 44 45 46 47 48 49
59 58 57 56 55 54 53 52 51 50
60 61 62 63 64 65 66 67 68 69
70

Figure 1. How the West was One + Three x Four

The game generates much enthusiasm and involves a fair amount of arithmetic. It has been played extensively by students from fourth grade through community college.

It would be much more difficult to use this game without a computer: the rules are sufficiently complicated and difficult to satisfy that some sort of referee would have to be available in many cases. It is important that errors in computation, such as 2+3x5=25, be pointed out immediately to the student. In addition, the computer adjusts the frequency with which different numbers appear on the spinners in order to make the game more interesting.

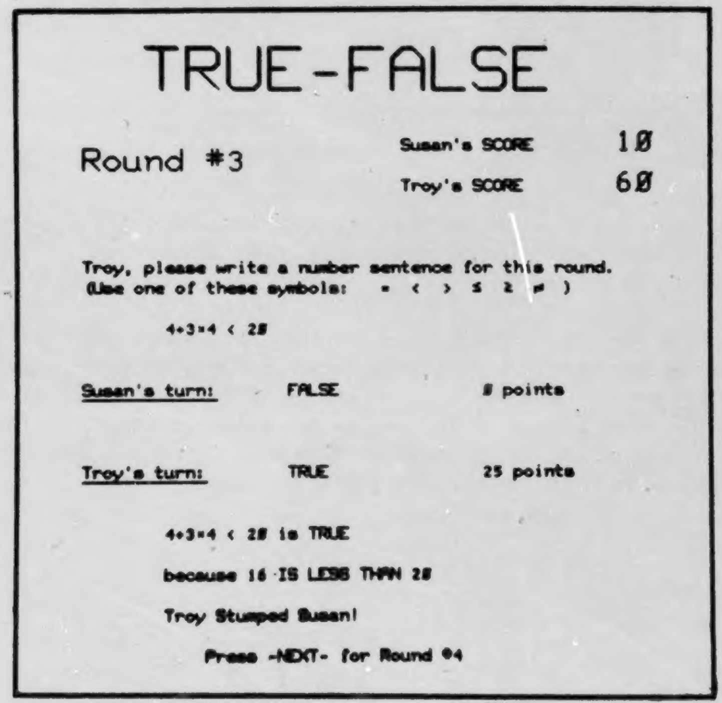
A computer can also provide a student with some of the special help he receives when his teacher watches him play the game. Resnick in her thesis (1975) observed several children playing "..West.." and classified their patterns of play. She demonstrated that students playing against the computer generally chose a simple strategy at the beginning and modified it very little after that. In other words, her work seems to indicate that students may obtain only limited benefit from educational games without outside intervention.

Brown (1976), starting from a suggestion of Resnick, wrote a program that evaluated the skill with which students played, kept track of patterns being used, and made appropriate suggestions for the improvement of play. The computer thus became an advisor and was able to combine direct instruction with the motivating game situation.

In much the same way that a teacher might intervene at appropriate moments, the computer can observe the student's game playing, correct conceptual errors, and make suggestions of strategy. Such a computer mentor can greatly enhance learning with a game played against a computer. It can be even more valuable when two students play together and discuss the computer's suggestions.

(2) True - False

"True-False" is a very simple game which makes good use of the computer and requires students to look at each other's work. Two students take turns writing down equations for inequalities (containing no variables). Each player individually guesses whether the equation is true or false and gains points when his guess is correct. (Figure 2)



```
TRUE-FALSE

Round #3                    Susan's SCORE    10
                            Troy's SCORE     60

Troy, please write a number sentence for this round.
(Use one of these symbols:  =  <  >  ≤  ≥  ≠ )

        4+3×4 < 20

Susan's turn:     FALSE              0 points

Troy's turn:      TRUE               25 points

        4+3×4 < 20 is TRUE
        because 16 IS LESS THAN 20
        Troy Stumped Susan!
            Press -NEXT- for Round #4
```

Figure 2. True - False

The player is encouraged to generate complicated and tricky equations in order to stump the opponent. However, he must know whether his equation is true or false in order to gain points with his guess. When either player misjudges an equation, the computer evaluates each side of the equation and shows why the student was incorrect. This ability of the computer to evaluate the truth of the equations makes it possible for the game to be played without numerous arguments.

The game may be modified to require the use of special quantities in the equations. For example, trigonometric functions, fractions, or signed numbers may be required.

INDIRECT INTERACTION

Computer Storage of Student Work

It is often useful to have students work on a problem together. Each student is able not only to make his own contribution to the solution, but also to learn from another's work, to build on it, and to incorporate others' ideas into his work. Often this cooperative situation can be achieved with a computer when two students work together at a terminal. At times, however, it is desirable to ensure that each student work alone and that he solve a problem or generate a new solution without direct help.

The ability of computers to store data and to display it readily is very useful in providing activities of this type. Through a computer lesson, a student can study the ideas of other students that have been saved and then add his own contribution to the list.

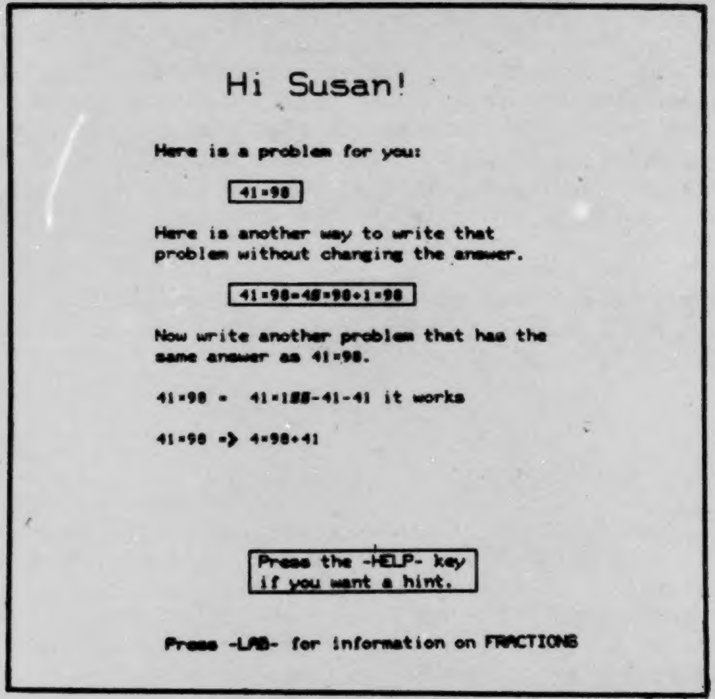
We describe three such activities that we have programmed.

(1) Problems

On a given day a particular arithmetic problem, such as 25x37, is chosen by the computer. Instead of being asked simply to solve the problem, the students are asked to write another problem which has the same answer. The computer supplies some examples and also keeps a list of the problems written by other students. (Figure 3)

After using this program for a period of time, students tend to expand their repertoire of techniques for modifying problems. Techniques introduced by one student tend to propagate through the class and appear in solutions written by other children.

The technique of having students cooperatively generate lists of solutions to such problems has been used successfully by many teachers without computers; the computer is also able to carry out such an activity efficiently.



Hi Susan!

Here is a problem for you:

41×98

Here is another way to write that
problem without changing the answer.

41×98=40×98+1×98

Now write another problem that has the
same answer as 41×98.

41×98 = 41×100-41-41 it works

41×98 => 4×98+41

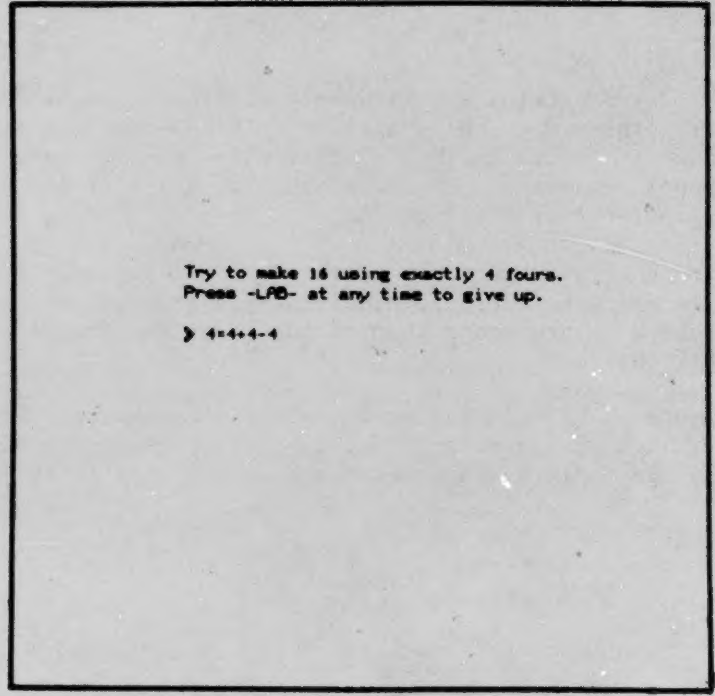
Press the -HELP- key
if you want a hint.

Press -LAB- for information on FRACTIONS

Figure 3. Problems

(2) Four 4's

Robert Wirtz (1974) has given the following problem: For each number from 1 to 100, write an arithmetic expression with that value, containing exactly four 4's and no other digits. (Figure 4)



Try to make 16 using exactly 4 fours.
Press -LAB- at any time to give up.

> 4×4+4-4

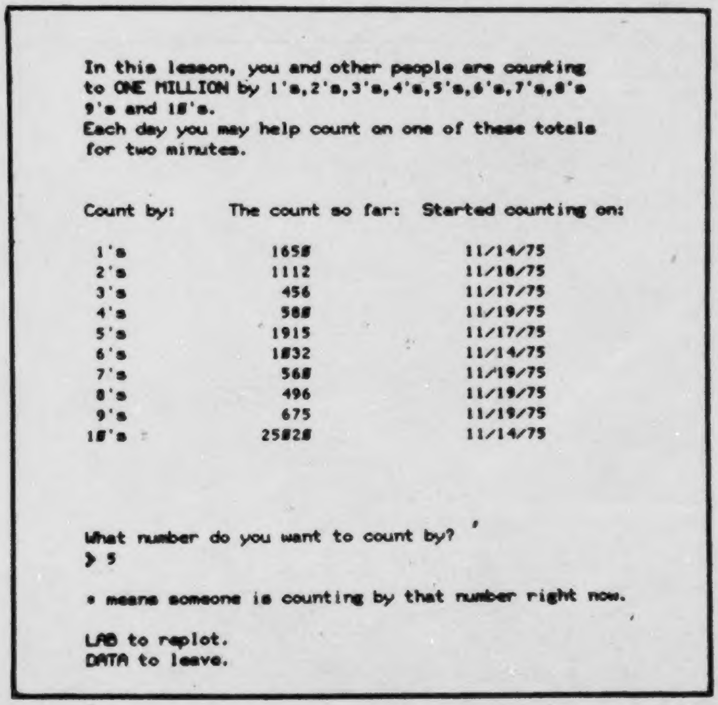
Figure 4. Four 4's

For example, the number 1 can be written as (4/4)x(4/4) and 10 is (44/4.4).

The problem is rather difficult and is reasonably done as a group activity carried out over an extended period of time. Our computer program allows a student to participate in the process of generating a solution to the problem and also to see the contributions of others. The computer determines whether an expression for a particular number is correct and provides information to the student on how difficult it is to generate solutions for different numbers.

(3) One Million

Counting to a million by steps of size 1 to size 10 is an activity which is certainly not mathematically complex. However, it provides a certain amount of satisfaction as well as some understanding of the magnitude of numbers. It is also an activity which is very tedious when done individually. We provide a program which allows students to skip-count to one million cooperatively over a period of time. (Figure 5)



```
In this lesson, you and other people are counting
to ONE MILLION by 1's,2's,3's,4's,5's,6's,7's,8's
9's and 10's.
Each day you may help count on one of these totals
for two minutes.

Count by:     The count so far:  Started counting on:

 1's                1650             11/14/75
 2's                1112             11/18/75
 3's                 456             11/17/75
 4's                 580             11/19/75
 5's                1915             11/17/75
 6's                1032             11/14/75
 7's                 560             11/19/75
 8's                 496             11/19/75
 9's                 675             11/19/75
10's               25020             11/14/75



What number do you want to count by?
> 5

* means someone is counting by that number right now.

LAB to replot.
DATA to leave.
```

Figure 5. One Million

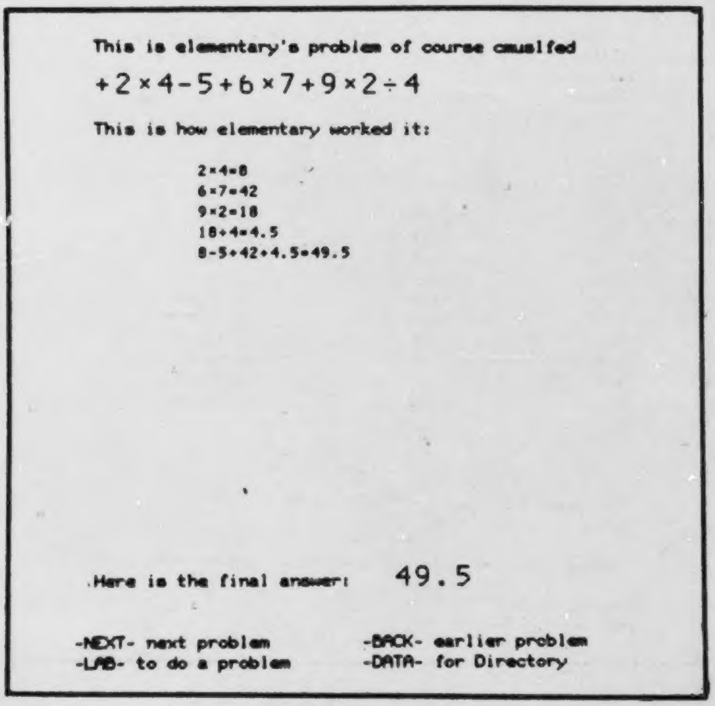
The computer keeps 10 separate tallies for skip counting by increments of 1 to 10. Students soon notice that certain numbers such as 1,2,5 and 10 are chosen more frequently by their fellow students and that these numbers are easier to count by.

Students USE Other Students' Work

Collecting and displaying student work serves three purposes. It boosts the ego of the student who did it, it provides ideas to the other students who look at it, and it provides the teacher with information on the level of the student's performance. We describe two such display programs.

(1) Your Way

Students pose arithmetic problems and then write in words and symbols the method they used to work the problem (Figure 6). The students' problems and solutions are kept by the computer for other students to see. A special feature of the program allows students to check computations by writing equations which the computer then judges to be true or false.



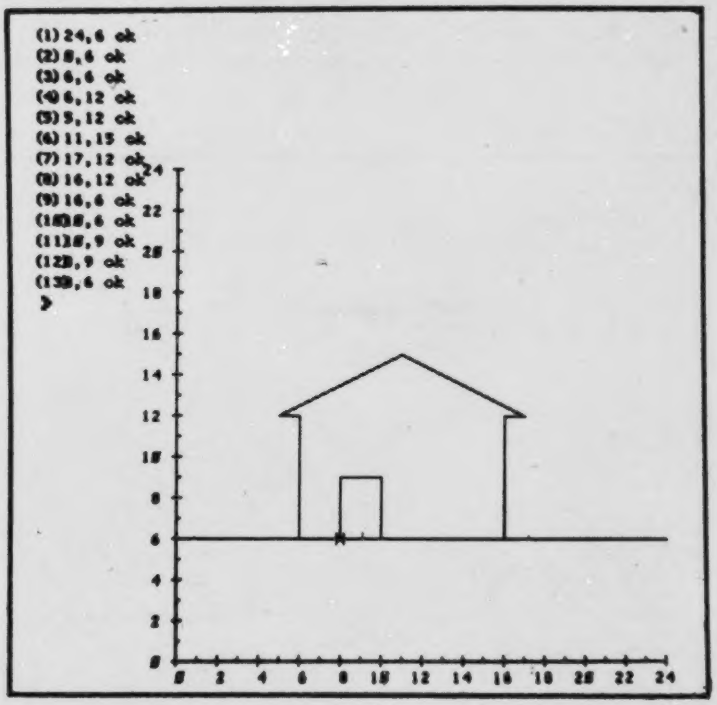
```
This is elementary's problem of course cmuslfed
+2×4-5+6×7+9×2÷4

This is how elementary worked it:

          2×4=8
          6×7=42
          9×2=18
          18÷4=4.5
          8-5+42+4.5=49.5

Here is the final answer:   49.5

-NEXT- next problem          -BACK- earlier problem
-LAB- to do a problem        -DATA- for Directory
```

Figure 6. Your Way

Examination of student work saved by this program over the past two years is not encouraging. The number of students who were able to describe in any detail how they worked a problem and the number of problems that were worked in an interesting way are equally small. This may be in part a reflection of the small amount of work of this type which students do in their classrooms. It is clear that the program as it now stands does very little to improve the student's proficiency in this area. It is perhaps possible that a more structured program which restricts students to a few pre-selected problems and provides several examples of solutions would improve the situation.

(2) Dot-to-Dot

As practice in plotting points on the Cartesian coordinate system, a student creates his own picture on the computer screen by specifying the end points of line segments (Figure 7). When finished, the student gives his picture a title and stores it in the program's picture library. Other students may then choose this title from the library and reconstruct the picture.



```
(1)24,6 ok
(2)0,6 ok
(3)6,6 ok
(4)6,12 ok
(5)5,12 ok
(6)11,15 ok
(7)17,12 ok
(8)16,12 ok
(9)16,6 ok
(10)10,6 ok
(11)10,9 ok
(12)8,9 ok
(13)8,6 ok
>
```

Figure 7. Dot-to-Dot

To reconstruct a stored picture, a student is given the coordinates of the points. He must locate each point by touching the appropriate location on the computer screen's graph. As the points are plotted correctly, the picture appears.

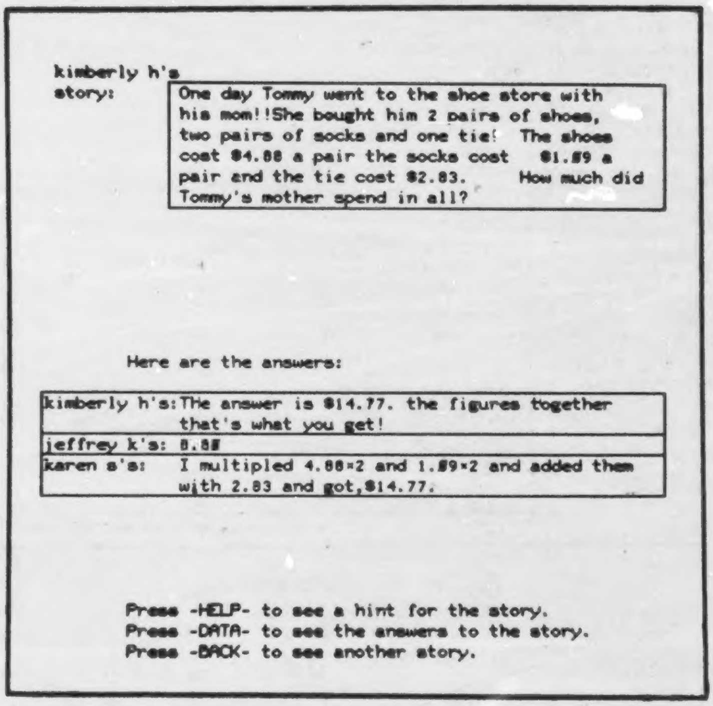
The picture library activity gives students an opportunity to be creative and to make puzzles for other student's use, while developing their mathematical skills. The program encourages students to explore the coordinate system in order to make pictures that are challenging and appealing to fellow students.

EXHANGING PROBLEMS AND SOLUTIONS

The programs we have described up to now make direct use of the computer in handling the mathematical content of the material being taught. The programs are written to make judgments on the accuracy of solutions and to perform calculations. In the case of student-generated word problems it is essentially impossible for a computer program to make any decisions about the mathematics. In spite of this, the computer can be programmed to encourage students to exchange word problems and work them.

Math Stories that are......By Kids

"By Kids" provides a mechanism for entering word problems into the machine (Figure 8). The student who writes the story problem also provides up to three hints for his problem and writes what he thinks is the solution. Once the student is finished, other students read his problem and the hints he provided, and then write their answers to the problem. After this, they may see the list of other people's solutions.



```
kimberly h's
story:        One day Tommy went to the shoe store with
              his mom!!She bought him 2 pairs of shoes,
              two pairs of socks and one tie!  The shoes
              cost $4.88 a pair the socks cost   $1.09 a
              pair and the tie cost $2.83.    How much did
              Tommy's mother spend in all?


       Here are the answers:

kimberly h's:The answer is $14.77. the figures together
              that's what you get!
jeffrey k's: 8.80
karen s's:    I multipled 4.88×2 and 1.09×2 and added them
              with 2.83 and got,$14.77.


       Press -HELP- to see a hint for the story.
       Press -DATA- to see the answers to the story.
       Press -BACK- to see another story.
```

Figure 8. Math Stories that are ....By Kids

Student use of this lesson has been very encouraging. The range of problems has been much broader than that found in texts. The students have obviously been very interested in writing problems and responding to them. In many cases, the proposer of the problem supplied an incorrect answer. Other times students misinterpreted the wording of the problem. The important thing is that the program makes it very easy for students to engage in the activities of writing problems, finding solutions, and seeing the ways in which others respond to them.

STUDENT-TEACHER COMMUNICATION

No computer-based instructional system can replace human interaction. In the case of our instructional system it has been important for us to communicate with the student users of our programs. Our lessons were being used by over 300 students during school year 1975-76; it was impossible for us to see all of them even once a day. There were programming errors which caused the lessons to malfunction. There were problems in understanding the content and places where the presentation was unclear. There were questions which we had not forseen. There were times when personal messages needed to be exchanged.

We therefore provided a program for exchanging notes about programs; it was available to students at the end of each program they executed We were able to read the notes at our convenience and write responses to the students, who read them the next time they used the computer.

The notes program was invaluable. Students found many minor errors in programs. They made suggestions for improvements in many lessons. They asked questions about topics that were not covered. They made general comments about the way they felt about instruction via computer. They were communicating, in writing, about mathematics.

We learned which approaches were successful and which were not. Many students became more aware, involved, and critical of their instruction because what they wrote often affected the instructional package.

Classroom teachers are in a much better position to communicate with students than we are, but they can also benefit from the Student Notes communication program. A teacher does not need to interrupt current activities to talk with students who are using the computer terminal. Instead, the students can write what they want when they want, and the teacher can respond at a time that is convenient.

Similar student notes programs have been used by several other curriculum development projects on the PLATO system; the feature is now provided by the system.

We have described several ways in which computers can be used to foster student communication. There is, however, a more fundamental way in which computers can help in this endeavor as they change the role of the classroom teacher.

Today the teacher is heavily involved in the process of presenting factual material to students and checking their understanding. The teacher must also perform many administrative and managerial tasks. As the use of computers in the classroom increases, the teacher will be freed from many of these activities and be able to spend more time and energy in direct communication with students. He will be able to arrange more frequent and more meaningful student-group activities and help students develop communications skills.

We look forward to the increasing use of computers as a tool for humanizing the process of education.

ACKNOWLEDGMENTS

The authors gratefully wish to acknowledge the efforts of Paul Cohen, Brand Fortner, Mike Hightower, Jeff Peterson, Peter Steinberg, Jim Wilmot, and Bob Zielinski, who contributed to the programming of these lessons.

REFERENCES

1. Brown, John Seely and Burton, Richard R., "A Tutoring and Student Modelling Paradigm for Gaming Environments," Proceedings for the Symposium on Computer Science and Education, Irvine, California, February, 1976.

2. Resnick, Cecily Ann, "Computational Models of Learners for Computer Assisted Learning," Doctoral Dissertation, University of Illinois at Urbana-Champaign, 1975.

3. Seiler, Bonnie A. and Weaver, Charles S. "Description of PLATO Whole Number Arithmetic Lessons," CERL Publication, 1976.

4. Wirtz, Robert, Mathematics for Everyone, Curriculum Development Association, 1974, p. 1.

CASPER: An Interactive Approach To Computer-Managed Testing

Charles Guerra
Harlan McDaniel
Joseph Kaufman
David Fuller

The object of this mode of instruction is to free the student from time restrictions as much as possible, so the student can progress through the curriculum at whatever rate best suits the individual's study habits, rather than at a rate determined by any external criteria. A student starts a self-paced course when he feels ready. The course is generally segmented into discrete modules, or units of instruction, each with stated objectives--concepts that are to be assimilated or skills that are to be acquired through study of the material. The substance of the units can consist of assigned readings, a taped lecture, problems from a workbook, a slide show, film loops, a computer-assisted instruction lesson, or any combination of these, just to mention a few. When a student has finished studying a particular unit, it is generally necessary to complete some assigned task to demonstrate mastery of the material. This might consist of some problems to be worked, an oral examination by the instructor, or a written test. Most courses also have examinations testing the student's cumulative understanding of several successive units. There is no schedule determining when students must take these tests; each student works at his own rate and takes each test when he feels ready. It's normal to observe students taking tests over the same material at different times.

From the description given above, it is clear that self-paced courses often include a large number of tests. Moreover, there are many self-paced courses. In fact, the College offers self-paced versions of every course in its core curriculum--the series of courses that every pharmacy student must take, regardless of field of specialization. This curriculum consists of 33 courses distributed throughout four years of professional study.

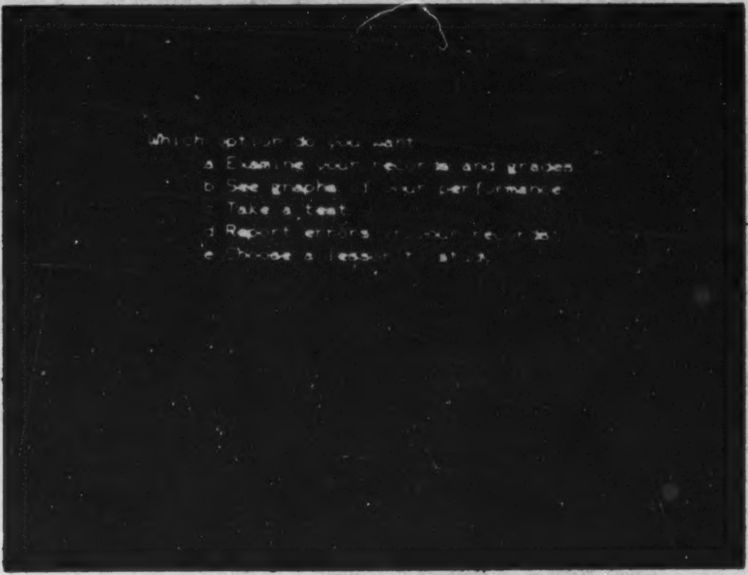
With all these tests to be administered in all these courses, the idea of automating this process is bound to arise. In addition, a lot of the concomitant bookkeeping can be organized more efficiently and accurately by a computer. Therefore, the College of Pharmacy decided to develop a computerized facility to expedite testing and record-keeping.

The University of Illinois' PLATO IV[1] was chosen as the system on which to establish the new facility. Twelve PLATO terminals were obtained, 10 of which are concentrated in a special room of the Pharmacy building (locally referred to as "The PLATO room").

The result that emerged from all this is a data-management system called the Computer-Aided Student Progress Evaluator--CASPER. CASPER is entrusted with all the data concerning self-paced students and courses. The details of CASPER's record-keeping capabilities have been delineated elsewhere. The purpose of the present paper is to describe in detail one facet of the system: CASPER's test administration capabilities.

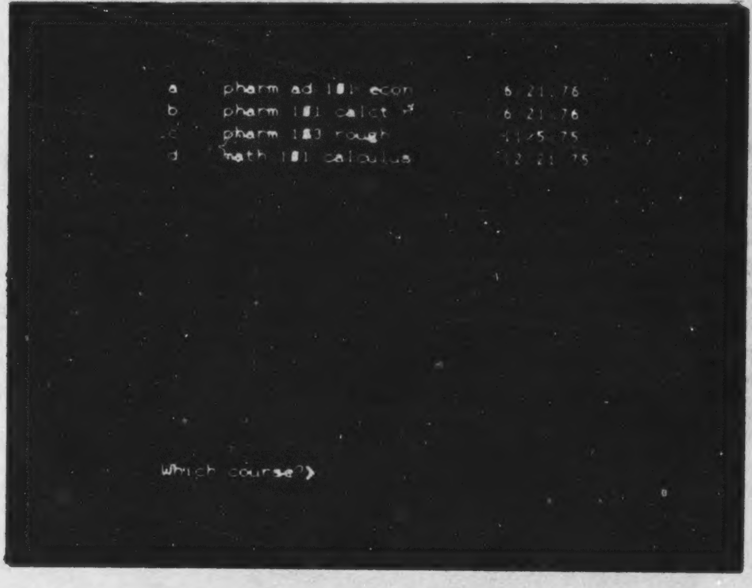
The nature of testing through CASPER flows from the nature of self-pacing. The most obvious manifestation of this is that CASPER is ready when the student is. A student can come to the PLATO room any time during normal business hours when ready to take a test. Then, all the student has to do is identify himself (by signing onto the PLATO system), indicate in which course he wants to take a test, and specify on what unit of the course he wants to be tested (figures 1, 2 and 3). Since the instructor is not present, CASPER takes responsibility for preventing the student from doing anything that violates the rules of the course, such as retaking a test to improve his grade. CASPER also takes responsibility for warning the student if he has chosen to do something that could cause him trouble, such as taking a test without having completed its prerequisite. The rules of the course are specified at the time the tests are put into CASPER (figure 4).

Testing on CASPER differs little in its essentials from testing on PAPER: tests consist of questions; each question has a set of possible responses designated "correct"; and correct responses must be made to a certain number of questions to pass the test. But the upright plasma panel is substituted for the flat desktop, and the PLATO IV keyset for the #2. These mold the character of test administration under CASPER. CASPER presents the questions one at a time, displaying the test of the question upon the screen and inviting the student to type in an answer. The student need not answer a question immediately, the first time it is presented; he can skip it and come back to it later. By pressing certain keys, the student can go forward to the next question, return to the previous question, or access a display that allows him to specify which question of the test he would like to go to. After the student has seen each question once, CASPER reminds him if he has left any questions unanswered, and then presents these unanswered questions again. This makes it impossible for a student to inadvertantly leave a question unanswered since CASPER will continue to do this until the student has given a gradable response to each question on the test. Then, when there are no more unanswered questions, the student may take a last look at the test. Each question is shown again, along with the response last given, so that the student can change his answer, if he so desired (figure 5). The student can skip around from one question in the test to another as much as he wants, or end the review at any time. The test can be thus reviewed as many times as desired, until the student is completely satisfied that every question has the response he intends.



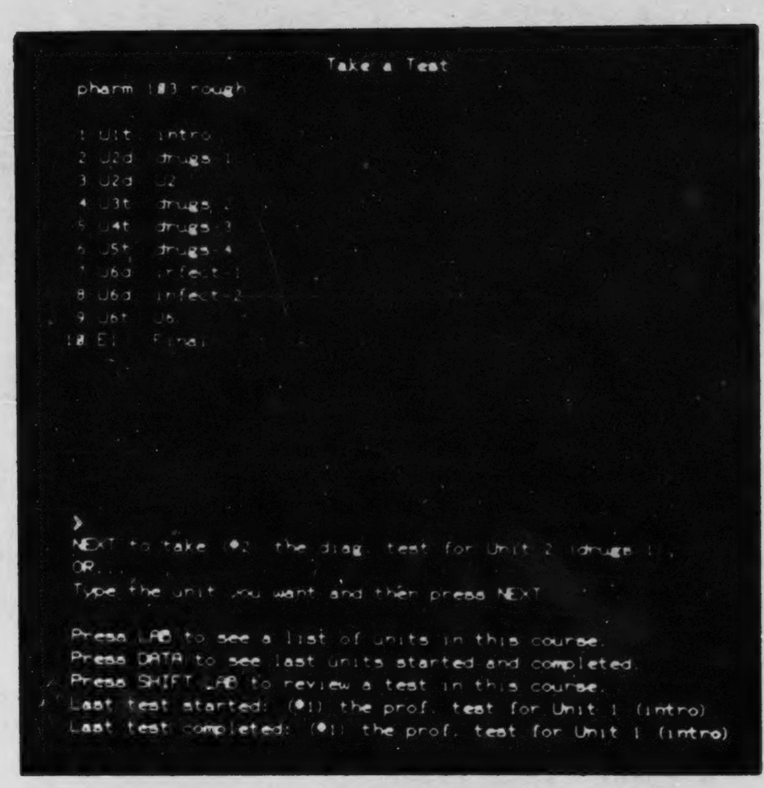
```
Which option do you want
      a Examine your records and grades
      b See graphs of your performance
      c Take a test
      d Report errors in your records
      e Choose a lesson to study
```

Figure 1. Main options page display as seen by student. By pressing the key for the letter to the left the system completes the action to its right. The student presses c to take a test.



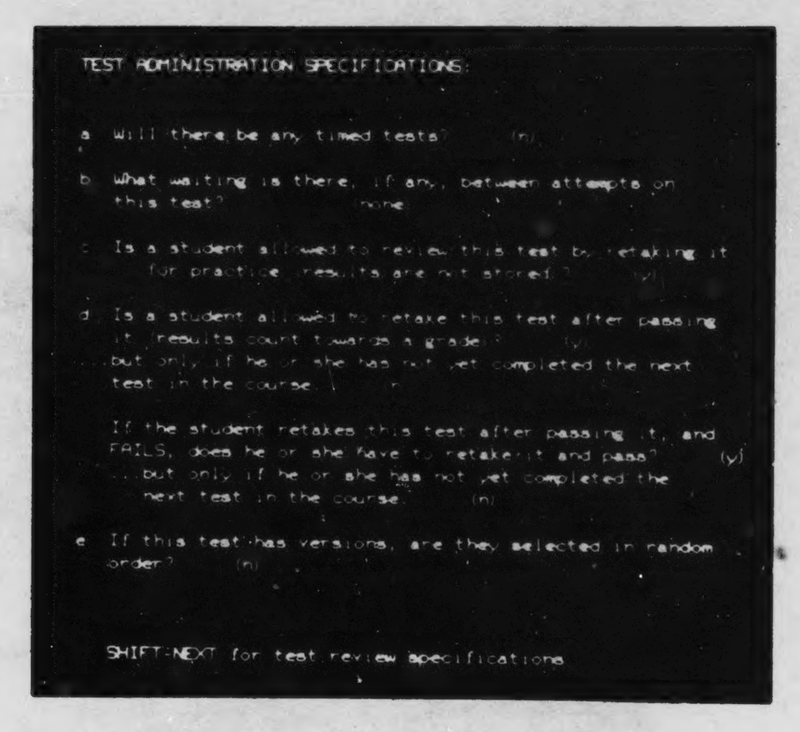
```
   a    pharm ad 101 econ       6/21/76
   b    pharm 101 calc          6/21/76
   c    pharm 103 rough        11/25/75
   d    math 101 calculus      12/21/75






Which course?>
```

Figure 2. Course list display as seen by student. The department name, course number, informal name and start date for courses in which the student viewing this display is registered. Typing a, b, c or d selects the course in which the student wants to take or review a test.



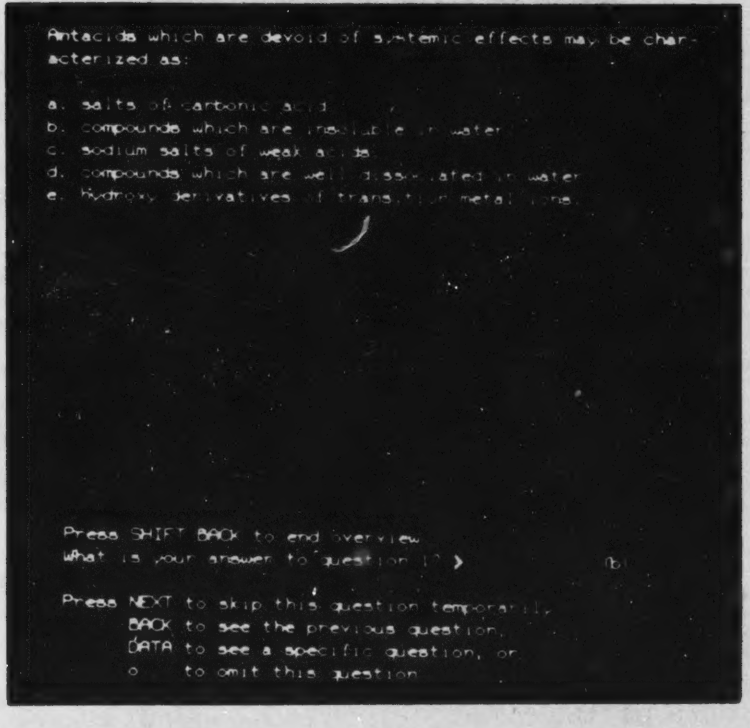
```
                        Take a Test
  pharm 103 rough

  1 U1t  intro
  2 U2d  drugs 1
  3 U2d  U2
  4 U3t  drugs 2
  5 U4t  drugs 3
  6 U5t  drugs 4
  7 U6d  infect-1
  8 U6d  infect-2
  9 U6t  U6
 10 E1   Final





 >
 NEXT to take (#2) the diag. test for Unit 2 (drugs 1),
 OR...
 Type the unit you want and then press NEXT

 Press LAB to see a list of units in this course.
 Press DATA to see last units started and completed.
 Press SHIFT-LAB to review a test in this course.
 Last test started: (#1) the prof. test for Unit 1 (intro)
 Last test completed: (#1) the prof. test for Unit 1 (intro)
```

Figure 3. Unit list display as seen by student after selecting course Pharm. 103 with informal name "rough". The name of unit (U for unit, E for exam), the unit number, the unit test type (d for diagnostic, t for proficiency test) and the test's informal name are listed. CASPER indicates the first unit for which the student has met the prerequisites (number 2 in this example). CASPER also allows the student to override its recommendation. (On a later display the proctor is warned if this has occurred.) Since the use of units display and the last units started and completed display are optional, CASPER reminds the student of the current function of the LAB and DATA keys. The student may also elect to review a previously taken test by holding down the SHIFT key and pressing the LAB key.



```
TEST ADMINISTRATION SPECIFICATIONS:

a  Will there be any timed tests?        (n)

b  What waiting is there, if any, between attempts on
   this test?              none

c  Is a student allowed to review this test by retaking it
      for practice (results are not stored)?        (y)

d  Is a student allowed to retake this test after passing
   it (results count towards a grade)?       (y)
  ...but only if he or she has not yet completed the next
   test in the course.         (n)

   If the student retakes this test after passing it, and
   FAILS, does he or she have to retake it and pass?       (y)
    ...but only if he or she has not yet completed the
       next test in the course.        (n)

e  If this test has versions, are they selected in random
   order?      (n)


    SHIFT-NEXT for test review specifications
```

Figure 4. A test administration specifications display as seen by a clerk entering a test. This display contains a few of the testing procedure options available to an instructor. The options selected indicate that (a) the test is not timed; (b) there is no waiting time imposed between attempts of the test; (c) the student may retake the test after pasing it (to either study the questions or improve his grade) and if the student fails that test, he must retake it; and (e) the student will be assigned a version of the test in the order listed on CASPER.

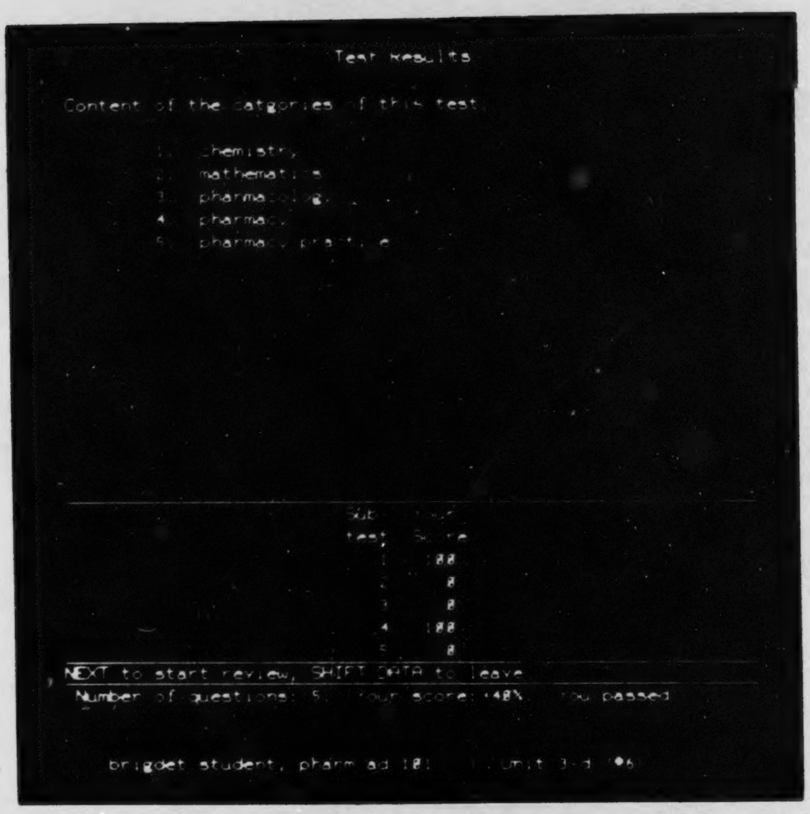


```
Antacids which are devoid of systemic effects may be char-
acterized as:

a. salts of carbonic acid
b. compounds which are insoluble in water
c. sodium salts of weak acids
d. compounds which are well dissociated in water
e. hydroxy derivatives of transition metal ions

Press SHIFT BACK to end overview
What is your answer to question 1? >               (b)

Press NEXT to skip this question temporarily,
      BACK to see the previous question,
      DATA to see a specific question, or
      o    to omit this question
```

Figure 5. Question display as seen by student during review of the test before grading. The review may be terminated at any time by pressing the keys SHIFT-BACK. The student previously answered b and may now change the answer. Alternatively, the student may leave the answer the same by pressing the NEXT key for the next question, the BACK key for the previous question or the DATA key to select a specific question. If the student does not want to respond to the question, he would type o, for omit.

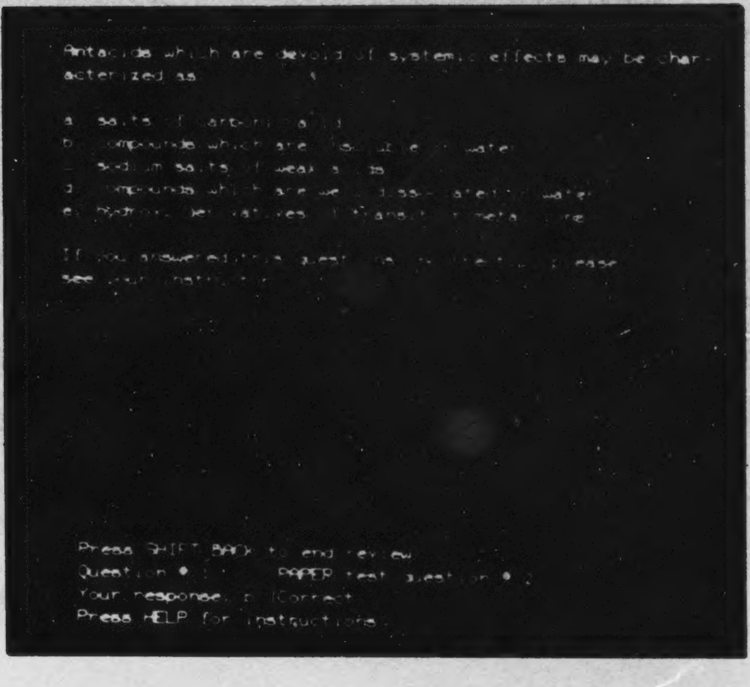
Then, the student has only to press a special combination of keys to signify completion, and CASPER will grade the test immediately. This has proven to be the most popular feature of the automated testing system, since the student discovers at once whether he passed and what his score is. On a different medium, this can be separated by hours or even days from the completion of a test. If the test is divided into sections, or subtests, the student is also shown how he did in each part of the test (figure 6). After the test results have been displayed, the student can review the test, either immediately or at a later date. While reviewing, the student can have CASPER show either all the questions on the test, or just the ones marked wrong. For each question presented CASPER displays the actual question, the student's answer to the question, the correct answer to the question, and any diagnostic remarks pertinent to the question (figure 7). This diagnostic comment is displayed directly underneath the text of the original question when the student looks at the item while reviewing the completed test. It usually serves to explain why the correct answer is correct, or to suggest someplace to refer for an explanation of the concepts involved. Diagnostic comments are appended to their respective questions at the time the test is being typed into CASPER (figure 8). When used appropriately, diagnostic comments can greatly increase the value of the post-test review. In general, this test review section serves to help the student assess his performance on the test and his understanding of the material it covers. When, at last, the student is done with this final review, he has completed the process of taking a test.

In a self-paced course, a student must demonstrate proficiency in the material covered by a given unit of instruction before he is allowed to move on to any units for which it is a prerequisite. In practice, this means that when the unit has a test associated with it, if the student does not achieve a passing score, he takes the test again until he finally passes it. This requires that some provision be made for a student to be able to to test himself in a unit more than once without taking the same test as he took on one of his previous attempts. There are a couple of different approaches that are used to tackle this problem. One approach is to have a relatively large pool of questions associated with each test and, every time a student takes the test, to select a small number of them at random for the student to answer. Another approach is to associate two or three different tests on the same material with the unit, one of which is to be selected to administer each time someone wants to take a test in the unit. CASPER can handle both of these approaches and any combination of them.



```
                         Test Results

Content of the catgories of this test:

          1.  chemistry
          2.  mathematics
          3.  pharmacology
          4.  pharmacy
          5.  pharmacy practice




                           Sub-  Your
                           test  Score
                              1   100
                              2     0
                              3     0
                              4   100
                              5     0
NEXT to start review, SHIFT DATA to leave
 Number of questions: 5   Your score: 40%   You passed

     brigdet student, pharm ad 101    unit 3-d (06)
```

Figure. 6. Test results display as seen by a student. This display has room for an optional 20 line comment in addition to listing the test scores. In this example, the comment lists the contents of each subtest, or category. The comment is also used to warn the student of special administrative rules, such as a 12 hour delay between re-takes of tests. The subtest scores inform the student of how well he did on each part of the test. The total number of questions, total score and the you passed/failed message quickly tell the student the results on the test. The bottom line of student identification is used on a CASPER system lacking a printing terminal. In that case, the score may be manually written down for later transmission of daily results to the instructor. In our system, a GE TermiNet prints out test logs at the end of each day.



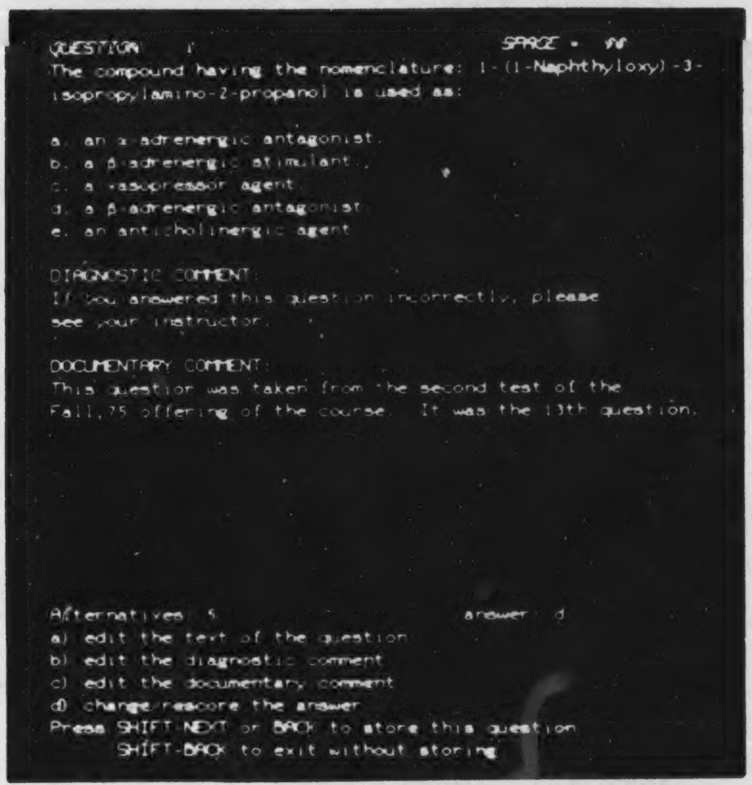
Antacids which are devoid of systemic effects may be characterized as

a. salts of carbonic acid
b. compounds which are insoluble in water
c. sodium salts of weak acids
d. compounds which are well dissociated in water
e. hydroxy derivatives of transition metals are

If you answered this question [illegible], please see your instructor.

Press SHIFT BACK to end review.
Question # 1   PAPER test question # 2
Your response: b (Correct [illegible]
Press HELP for instructions.

Figure 7. A missed question display with diagnostic comment as seen by a student after the test has been graded. The top six lines of text are the original question. The following two lines are a short diagnostic comment - normally an explanation would appear here. The text below tells how to end the review of the test, what the question number was, what number that question is identified with on the instructor's (paper) master list of questions, what his response was and whether it was correct.



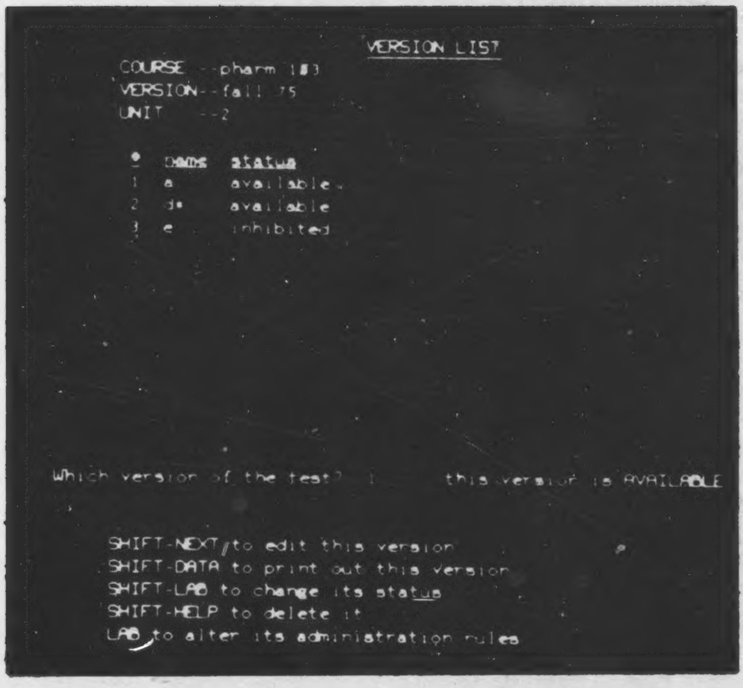
```
QUESTION:  1                               SPACE = 98
The compound having the nomenclature: 1-(1-Naphthyloxy)-3-
isopropylamino-2-propanol is used as:

a. an α-adrenergic antagonist.
b. a β-adrenergic stimulant.
c. a vasopressor agent.
d. a β-adrenergic antagonist.
e. an anticholinergic agent

DIAGNOSTIC COMMENT:
If you answered this question incorrectly, please
see your instructor.

DOCUMENTARY COMMENT:
This question was taken from the second test of the
Fall,75 offering of the course.  It was the 13th question.




Alternatives: 5                        answer: d
a) edit the text of the question
b) edit the diagnostic comment
c) edit the documentary comment
d) change/rescore the answer
Press SHIFT-NEXT or BACK to store this question
      SHIFT-BACK to exit without storing
```

Figure 8. Question entry display as seen by clerk entering question. Reading the display from the top we see that this is question number one and that there are 98 computer words of storage left for entry of new text for this question. The question text itself appears next followed by the diagnostic comment and documentary comment. The latter is only seen by the instructor. Next we see that there are 5 alternative answers, of which only answer d is correct . Pressing a, b, c, or d in this display causes that action to be taken which appears to the right of those letters at the base of the screen. Finally instructions are given on how to store the question in memory and how to avoid it.

Such flexibility is possible because of the format in which the tests are arranged. *Every unit test in CASPER* contains one or more "versions" of the test (figure 9). A verson consists of a pool of questions and a set of rules on how to administer the questions when giving the test to a student. Thus, a unit utilizing the selection-from-a-pool approach would consist of one version with a large number of questions, whereas a unit utilizing the other approach would consist of two or more versions, each having a moderate number of questions (figure 10). In the first case, the version administration rules would specify that a certain number of the questions are to be selected at random; in the second case, the version administration rules would specify that all of the questions are to be presented.

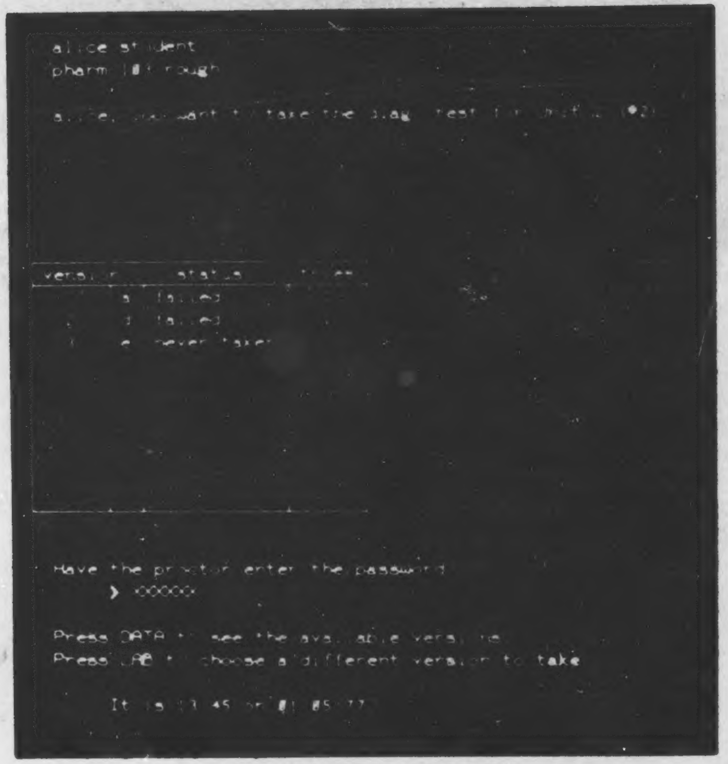
Sometimes adequate information about a student's total absorption of the material in a unit cannot be discerned from a single overall test score. Consider the case of a student who achieves a passing score on a test, even though he did poorly on questions pertaining to a particular concept, due to the fact that he did quite well on the rest of the test. This could signify an inadequate understanding of certain aspects of the material, which could prove deleterious in later study or practice. Thus, it is often important to have some means of measuring the student's grasp of each of the separate topics covered in a unit. The most common solution to this problem is to divide a test into subtests, or categories, such that questions are organized into distinct groups, each group having its own scoring rules and passing score. Then, if a test is sectioned according to topic, a student taking the test will get a score measuring his performance in each topic, as well as an overall score on the test as a whole, so that the student can more directly see where his strengths and weaknesses lie. This is one aspect in which CASPER far exceeds paper testing, since instructors and proctors dislike all the concomitant arithmetic of calculating several different test scores (and for that reason are inclined not to bother with categories). The use of categories has an additional advantage for tests administered by randomly selecting a certain number of questions from a large pool: it allows the mass of questions to be treated as several small pools, each with its own rules for question selection, instead of as one large pool. This assures that a sufficient number of questions are presented from each area of concern in the unit (figure 11).

Perhaps the most outstanding characteristic of CASPER is that it was basically designed by the faculty members interested in computerizing their tests. Most of these instructors had had experience with testing on PLATO through our previous testing facility.[3] Each instructor had suggestions for improvements and had additional demands for what PLATO would have to do in order to adequately administer testing in his courses. In constructing a facility that could satisfy all the diverse needs of the testing methods involved, a system has been



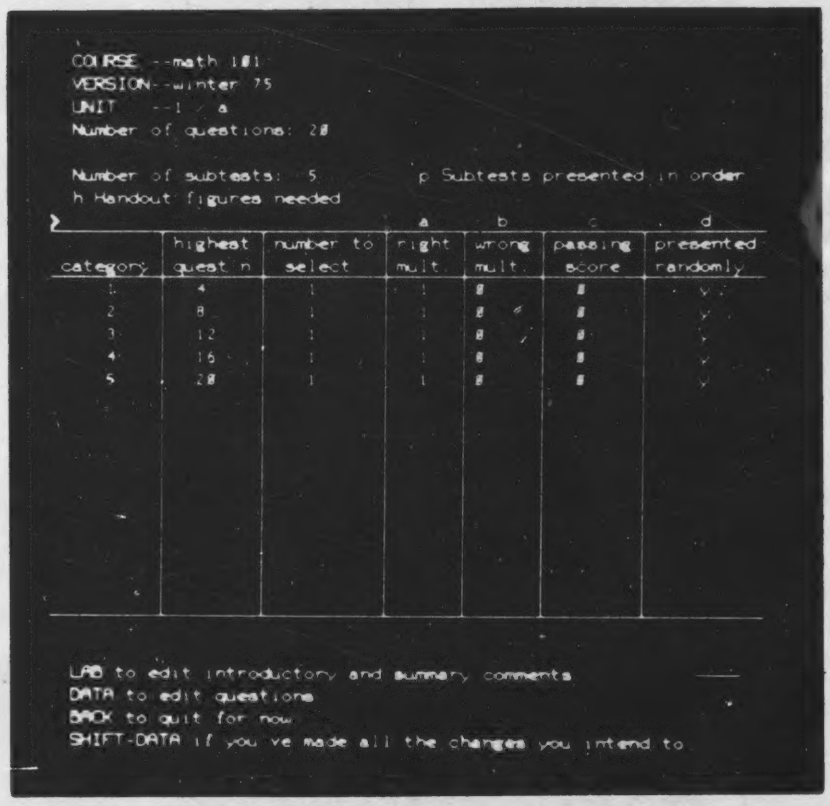
```
                         VERSION LIST
   COURSE  --pharm 103
   VERSION--fall 75
   UNIT    --2

    #  name  status
    1  a     available
    2  d*    available
    3  e     inhibited


Which version of the test?  1      this version is AVAILABLE

   SHIFT-NEXT to edit this version
   SHIFT-DATA to print out this version
   SHIFT-LAB to change its status
   SHIFT-HELP to delete it
   LAB to alter its administration rules
```

Figure 9. Version list display as seen by clerk. Reading from the top we see that the test involved is in the Pharm. 101 course that started in fall 1975. The test is in unit 2. There are three versions of this test but only version 1 and 2 (a and d) are available for student use. Version 3 (e) is inhibited because it is not yet complete. Pressing 1, 2 or 3 (or the corresponding letters) identified the test one wished to consider then pressing the keys identified (SHIFT - NEXT DATA, LAB and HELP) completes the action identified to their right.



```
alice student
pharm 103 rough

alice, you want to take the diag. test for unit 2 (#2)

version      status       taken
      1      failed
      2      failed
      3      never taken

Have the proctor enter the password
     > xxxxxx

Press DATA to see the available versions
Press LAB to choose a different version to take

     It is 13:45 on 01/05/77
```

Figure 10. Proctor security code entry display. Reading from the top, we see that a fictitious student named Alice Student is taking the course Pharm. 103, whose fictitious informal name is "rough". Alice has already selected the diagnostic test for unit 2 for administration. She has pressed the DATA key to show the versions of this unit diagnostic test available to her. She has failed both versions 1 and 2. Thus she will take version 3. She has called the proctor who viewed the screen, accepted Alice's choices and entered the secret code, seen by the student as a random number of x's. Had Alice desired she could have selected another version by pressing the LAB key. Alice viewed the display at 1:45 P.M. on 1/05/77.



```
COURSE --math 101
VERSION--winter 75
UNIT    --1 / a
Number of questions: 20

Number of subtests:  5          p Subtests presented in order
h Handout figures needed
>                                 a       b        c          d
          | highest | number to | right | wrong | passing | presented
category  | quest n |  select   | mult. | mult. |  score  | randomly
    1         4          1         1      0        0          y
    2         8          1         1      0        0          y
    3        12          1         1      0        0          y
    4        16          1         1      0        0          y
    5        20          1         1      0        0          y

LAB to edit introductory and summary comments
DATA to edit questions
BACK to quit for now
SHIFT-DATA if you ve made all the changes you intend to
```

Figure 11. Category description page. Reading from the top, the course is Math. 101, which started in winter of 1975. The test is version a of unit 1; there are 20 questions divided into 5 subtests. The subtests are to be presented in order and handout figures are used with the test. Pressing p or h changes the order to random or removes the reminder to the proctor to handout figures to students taking this test. Reading the entries in the table for category 2 we see that it contains questions number 5 through 8; only one of these four questions will be selected to be seen by the individual student. The student receives 1 point for each wrong answer and 0 points off for each wrong answer. The passing score is 0 and the questions in this category are presented randomly. The latter is vacuous since only one question from this category is selected for an individual student. Pressing the keys at bottom left of the display causes the corresponding action at bottom right.

produced that can satisfy the mode of operation in a wide variety of courses. Numerous options are available to an instructor enabling CASPER to follow his rules concerning every aspect of test administration: when to allow a student to take a test; how often a student can take tests; how to determine which version of the test a student should take; how to select the questions to be asked in the test; whether the questions should be presented sequentially or in random order; whether the student is allowed to change his answer once he has given a response to a question; whether the student is allowed to review the test after completing it; whether the student is permitted to come back some other time and review the test; and the list goes on.

The instructor also decides what measures PLATO should take to control access to his tests. CASPER offers six levels of protection ranging from a maximum that requires the student to present a written permit to the proctor on duty in the PLATO room, to a minimum that allows the student to take a test at any PLATO terminal on campus whether supervised or not. Most courses use an intermediate level of security, requiring the student to take tests in the PLATO room with a proctor present. The proctor verifies his or her presence by approaching the student's terminal and typing a secret password (figure 10). (Of course, CASPER does not display what the proctor is typing.) This password is generated algorithmically, so it can be made to automatically change every day, or even every hour, and the algorithm itself can be changed any time a proctor fears the password is known.

After an instructor has written his test and established his administrative rules, he is interested in the test results. Student scores on CASPER tests are reported to him daily via print outs of a test log. Further, the scores are stored on-line. The extensive on-line displays discussed in a previous paper[2] together with the testing system comprise the current CASPER computer managed instruction system.

Based on our experience with our current CASPER system, we plan several improvements. We are considering offering alternate types of questions, such as matching, numerical and short answer. Further test administration rules are planned. Finally, a complete test analysis package is under construction.

As a result of using CASPER the individual instructor is freed of the burden of administering and grading tests and is aided in evaluating his test questions.. The student benefits by being able to take tests at any time and receiving immediate feedback. The college benefits by having a central testing facility which can handle large numbers of tests in large numbers of courses, each with its own individual rules of test administration.

Footnotes

[1] Smith, Stanley G. and Sherwood, Bruce Arne, "Educational Uses of the PLATO Computer System," Science, 192, 344 (1976).

[2] McDaniel, H., Guerra, C., Kaufman, J., Bonetti, W., Galcher, W. and Lamprinos, P., "CASPER-The Record Keeping Component," Proceedings/1976 Summer Conference/ Association for the Development of Computer-Based Instructional Systems, Minneapolis, Minnesota, p. 195, 1976.

[3] H. McDaniel, D. Fuller, C. Guerra, J. Kaufman, P. Lamprinos, J. Long and C. Nelson, "CASPER I: The Testing Component," submitted to American Journal of Pharmaceutical Education.

CMI AND THE LOGICAL ANALYSIS OF CURRICULUM

Robert F. Yeager
Computer-based Education Research Laboratory
University of Illinois at Urbana-Champaign

The purpose of computer managed instruction (CMI) is to have a computer decide what a student is ready to study. The computer keeps track of the student's progress through a curriculum and makes prescriptions specifying what material the student should get next.

In order to do that job well, the computer needs three pieces of information: (1) a model of the subject matter; (2) data on the current state of the learner; and (3) a theory of instruction which ties the first two components together.

This paper is based on the experiences of the PLATO Elementary Reading group at the University of Illinois which tried to build a CMI system without that third component. The effort was finally abandoned after five years, and after considerable resources had been expended on trying to make the CMI system deliver reasonable prescriptions.

The Reading group placed the greatest emphasis on the first component of a CMI system: the model of the subject matter which was derived by making a logical analysis of the curriculum. This paper will explain what is meant by a "logical analysis of a curriculum"; how such an analysis was used as the basis of a CMI system; and what the effects of that approach were on the actual operation of the CMI system.

What Is a "Logical Analysis" of a Curriculum?

I have used the term "logical analysis" to mean much the same thing as "task analysis". In a task analysis, the main skills to be learned by the student are identified; and then those skills are broken down into their basic components. This is a deductive process in which the person making the task analysis starts with generalities and works backwards to the specifics. The goal of the task analysis is to produce a sequence of small steps which will lead the learner through the specifics to the generality; or form the basic components back to the main skill.

Typically a task analysis is done in the language of behavioral objectives. These state precisely what the student has to do to demonstrate that he has mastered a skill; and they include criteria which specify how the student's performance will be measured.

A logical analysis is a subset of a task analysis. There are three things which make a logical analysis a special type of task analysis. First, a logical analysis can be applied to larger goals. Task analyses are thought of as being applied to defined, limited objectives, such as, "adding single digit numbers which sum to a number less than ten." A logical analysis is meant to apply to larger goals, such as, "being a competent reader", which might take years to achieve.

Secondly, since it is more global, a logical analysis has to organize skills which are less closely related than the skills in a task analysis. A task analysis is usually thought of as merging a number of associated skills into one; task analyses are often described graphically as a pyramid with the sub-skills joined together by lines which lead to the main skill on top. A logical analysis, by comparison, has a more elongated, "strand" look; the lines do not begin to converge until very near the top; and at the lower levels, there are few lines which tie strands together.

For example, to be a competent reader, the student will have to learn phonetics, sight word skills, comprehension skills, and others. There are no logical connections among those skills; a student can progress far up the phonetics "strand" without having begun sight words or comprehension ; and vice versa.

The third and most important difference is that a logical analysis is restricted to the logical interconnections among skills. The only question that a logical analysis asks is whether skill B subsumes skill A; if so, they are marked as being connected. A task analysis, on the other hand, can include pedagogical and psychological considerations in determining whether to connect skills together.

For example, the ability to recognize letter names presumes the ability to visually differentiate letter forms; and so those two are connected. But the ability to recognize letter names is not required to learn phonics; some teachers successfully teach the sound that a letter makes before they teach the letter name. Nevertheless, many teachers would want to connect letter names to phonics for pedagogical reasons: it makes it easier for them to teach the letter sound if the teacher can refer to the letter by name. While such a connection is valid in a task analysis, it is not valid in a logical analysis.

Using a Logical Analysis in CMI

Translating a logical analysis of a curriculum into a computer-based curriculum management system is a computer programmer's delight. The curriculum designer's skills become the programmer's nodes; the curriculum hierarchy is treated as a tree structure in the computer science sense; and the criteria associated with each skill provide the variables used to decide if a node is completed. The computer, then, can maintain a map for each student showing how far that student has progressed in the curriculum; and it can use that map to generate prescriptions.

The Elementary Reading project used CMI to generate prescriptions for computer assisted instruction lessons. CAI lessons were associated with each skill; the CMI system determined which skill the student should work on and selected the appropriate lesson for that skill.

Each skill usually had several lessons associated with it; the lessons included diagnostic tests, instruction, and, sometimes, remedial instruction. The lessons were expected to tell the CMI system when the skill was completed; but the lesson could also tell the system to switch to another lesson within that skill. This was accomplished by requiring lessons to set a single "return value" which could be interpretted by the CMI system. For example, after a test, the test lesson would tell the CMI system either to mark the skill as completed or to queue up an instructional lesson for that skill.

The CMI System in Operation

The Elementary Reading CMI system actually appeared to operate fairly well. In 1974-1975 it kept track of about 200 students a day; and in 1975-1976 it selected lessons for over 300 students a day. Changes and additions were made to the system as problems developed. But eventually it became apparent the source of the problems was in the design of the CMI system itself insofar as that design was built around a logical analysis of the curriculum. The types of problems fall into three areas: (1) the quality of the prescriptions; (2) the effects of the CMI system on lesson development; and (3) the effects on implementation of the curriculum in classrooms.

Quality of the prescriptions. The CMI system was built on the assumption that the only sequence that was important was the logical sequence; thus, students should be able to work in several, unrelated skill areas (or strands) at the same time so long as they fulfilled the prerequisites for each area. This resulted in the "hit and run" curriculum in which students were bounced from one skill area to another within a single session. The assumption had been such diverse experiences would be reinforcing to the student, and would spread the instruction out over time. Instead it created confused students who could not explain in interviews what they did at the computer terminal.

Furthermore, while data showed that students interacted successfully with individual CAI lessons, other data indicated that students gained little or nothing from their overall exposure to CAI. One possible explanation of this paradox is that the sequence of instruction plays a critical part in the effectiveness of that instruction; and the sequence of instruction must be determined by more than the logical relationships among skills.

The problem of the "hit and run" curriculum was never solved. All of the changes that were tried were done in terms of the logical analysis; but what was needed was a redesign of the system to allow for pedagogical considerations in the selection of lessons.

Another flaw in the quality of the prescriptions was the system's propensity to get students stuck at certain skills. For one reason or another, a student might not be able to satisfy the criteria for a certain skill. When teachers are faced with this problem, they may employ any of several strategies ranging from dropping back several levels and strengthening the foundations, to going ahead anyway and looping back to pick up the troublesome skill later. The only thing that the CMI system could do was to grind away at the same skill until somehow it was passed. To do anything like a teacher would do would have required a redesign of the system to include pedagogical considerations.

The most ironic thing about the way the CMI system selected lessons was that, in spite of its sophistication, two students were often observed sitting side by side taking the same identical lesson. While no hard data have been gathered, it appears that the system did not individualize instruction in the way it was supposed to, and that students within the same classroom received approximately the same sequence of lessons!

Effects of Lesson Development. Designing lessons which were intended to fit into a logically structured CMI system meant that the lessons had to be focussed on a single skill. This had the positive effect of producing well defined lessons. But it also had the negative effect of excluding ideas for lessons which incorporated skills which either (a) were in different areas of the curriculum which were not logically connected; or (b) tried to synthesize several skill levels. Such lesson development was deferred until a way to work those lessons into the curriculum structure could be devised. Consequently most Elementary Reading lessons deal with single skills taught in isolation, and very few lessons attempt to integrate skills.

A second effect on lessons was that lessons gradually accumulated complicated algorithms to compensate for the inflexibility of the CMI system. Such algorithms eventually took up about ten percent of the overall lesson size; and they could add as much as fifty percent to the time it took to develop a lesson.

The algorithms were necessary to make the lessons more sensitive to pedagogical factors. The greatest need in lessons was to relate in idiosyncratic ways the skills taught in that lesson to other skills at the same level. For example, in teaching the letter name "m", it is helpful to know whether or not the student knows any letter names yet; and, if so, how well he knows them. The answers to those questions should determine the strategy used to teach "m".

Looking at the nature of most of the algorithms added on to lessons suggests that the most complicated part of CMI is correlating instruction at the same skill level; in other words, the most critical relationships in a curriculum are the horizontal ones which were dealt with by the add-on algorithms, not the vertical ones defined in the hierarchy of skills.

Effects on implementation. The teachers who used the Reading lessons never really understood the structure of the curriculum; this was true in spite of the fact that a 450 page teacher's manual had been prepared which described the lessons and showed how each lesson fit into the curriculum structure. Part of the reason for this may be that teachers operate in a richer context than what had been provided; the logical analysis of the curriculum did not provide them with the type of useful information they needed.

Special problems arose when teachers wanted to interface with the curriculum and to have students skip over skills. For example, a curriculum designer might arrange skills in a A, B, C order, but an individual teacher might want to see skill B presented first; if such input is allowed, the problem is which skill to present after B; skill A or skill C? And this seemingly simple problem becomes a conundrum if the skills are placed in a real context of a complete hierarchical tree with a cobweb of lines connecting the skills together.

Conclusion

A logical analysis of a curriculum may be a necessary condition for CMI, but it is definately not sufficient by itself. The experience of the Elementary Reading project shows that a CMI system must be able to include pedagogical and psychological variables in the algorithms used to prescribe lessons.

We should not confuse a theory of knowledge with a theory of instruction. The logical analysis of a curriculum constitutes a theory of the organization of knowledge within that subject matter; and, as such, it can be very informative. But computer managed instruction requires a theory of instruction which can analyze the current state of the student and prescribe the most effective instruction for him.

Reference

Risken, John and Ed Webber, "A Computer Based Curriculum Management System". Educational Technology Vol. XIV, No. 9 (September, 1974), pages 38-41.

A Minicomputer CAI System

by

Thomas T. Chen, Douglas W. Jones, A. B. Baskin,
Larry Sherman, and Allan H. Levy
Medical Computing Laboratory, School of Basic Medical Schiences
College of Medicine, University of Illinois

1. Introducation

A major portion of all computer based instructional material is written in the TUTOR [1] programming language supported on the PLATO IV [2] system developed by the Computer-based Education Research Laboratory (CERL) of the University of Illinois. The PLATO IV system and the associated TUTOR language have evolved together over the last 16 years and combine a number of powerful features that are not widely available in other systems. Though intended primarily for the delivery of instructional materials, the system has been used for various other programs that are facilitated by the interactive usage of a computer. PLATO IV, implemented on the CDC CYBER 70 series of computer, is capable of supporting 400 to 600 graphic PLATO IV terminals executing programs concurrently. The participation of thousands of programmers and users on the PLATO IV system has created an intellectual community with users from across the entire nation, and it is the establishment of that intellectual community which is perhaps PLATO's most significant accomplishment.

The large size of the PLATO IV system has given rise to significant problems in the management of system resources. In order to make the PLATO IV system more responsive to users, the system resources have been subdivided into a number of "virtual machines" called logical sites which are administered by site directors. Site (and course) directors control such system resources as disk space, sign-ons and access rights for users at a given site. The administration of system resources works well when the users of a particular site have uniform needs. However, when a quirk of geography forces users with widely differing system requirements to use the resources of a single site, resource allocation problems result. In addition to problems of resource allocation, the size of the PLATO IV system gives rise to problems in expanding the system. The use of a single centralized main frame computer with each terminal connected to it necessitates some form of telecommunication facility for remote users (with a resultant increase in cost for system usage). In addition, since the PLATO IV system depends on an "economy of size" to distribute the large initial cost of the central main frame computer over a large number of users, the replication of a PLATO IV system requires an extensive capital investment and the organization of a large number of initial users. (The use of a portion of a large main frame for a small

PLATO system solves the problems associated with size when the remaining system resources can be effectively utilized.)

This paper deals with the design and implementation of a minicomputer based system with many of the features of PLATO IV. This system supports the TUTOR language and can be used for the development of lesson material. Student access to individual lessons is provided in much the same way as on PLATO IV. Extensions to the programming system offered by PLATO IV include: enhanced data base management primitives, demand paging of the TUTOR program space, facilities for the development of modular systems of programs via run time linkage of individual programs, and the ability to use programming languages other than TUTOR. A large community of users can be formed by the creation of a depository network with many of these small systems tied to a central depository for the sharing of lesson materials and the supporting of on-line inter-user communication[3]. The computing system to be described below is the first node to be implemented in the proposed network and will be referred to as a "local node" in that network.

The local node is intended to provide the state of the art user interaction available on the PLATO IV system for up to 50 users. The system is constructed in such a way that the size of the system and the particular mix of system resources can be tailored to suit a specific group of users. Thus, the local node appears to constitute a readily manageable system allowing complete local administrative control and avoiding communication costs by serving only a highly localized group of users. The preliminary implementation results presented below indicate that a minicomputer is capable of supporting such a system without sacrificing major system capabilities.

2. System features

The PLATO IV system has convincingly demonstrated the benefits to be derived from a friendly and forgiving user interface to a computing system. Both the development and the use of programs on a computing system can be facilitated by a flexible user interface. The form of the system described here is closely patterned after that of the PLATO IV system. Every attempt has been made allow development and use of our system without extensive knowledge of its architecture.

2.1 Terminals

Unlike the PLATO IV system which uses a dedicated type of terminal and a specialized transmission code, the local node system supports both PLATO IV graphic terminals and alphanumeric CRT terminals. In both cases, the transmission code is a graphically oriented extension to ASCII providing compatibility with an industry standard terminal design. In many applications the CRT terminals can be used where graphic capability is not required, thus reducing the total cost of terminals for the system.

In addition to the actual terminal hardware, the system provides the line editing features which include: copy, shift-copy, erase, shift-erase, edit, and shift-edit. The presentation of a comment on a wrong answer and its erasure by the system when no longer needed are also supported. The terminal handling software described here is implemented in such a way that all of the features are available in any programming language by specialized input/output service calls.

2.2 System command language

Although not normally described as such, the PLATO IV student mode and author mode pages constitute the job control language for the PLATO IV system. By selectively arming the black control keys for the common functions and providing menus of less frequently used options, the user is allowed to specify commands to the system that allow him to move about in the system with great ease. Our system supports essentially the same author mode user interface as that supported on PLATO IV and in addition provides access to the vendor supplied job control language for selected users. The student mode page, although not yet implemented, is a simple subset of the capabilities of the author mode options. Unlike the PLATO system, our system utilizes a multilevel directoried filing system which is accessible to the user. Thus, on the author mode page a program name and the name of the directory in which it is found must be specified (if the directory is other than the default for the given user). Thus multiple copies of programs at various levels of development can be maintained simply by using separate directories.

2.3 System editors

A number of system level programs are provided to aid in the development of program materials. Several editors are provided for text file preparation. These include an editor patterned after the PLATO IV editor and an editor based on QED [6]. Taken together, the editors provide a text editing mechanism that even includes the ability to "program" complex file manipulations and specify the scope of the file manipulation. Special purpose editors for the preparation of complex graphic displays and the preparation of PLATO IV terminal alternate charactersets are provided.

2.4 Language Processors

Although the TUTOR language on the PLATO IV system is very powerful for CAI and CMI, it is desirable to support other general purpose and CAI languages for ease of program transportation. In our implementation FORTRAN and BASIC are supplied by the vendor, RATFOR (a block oriented FORTRAN extension) is available from Bell Laboratory, while LISP 1.6, PASCAL, PILOT and TUTOR have been developed locally.

The existence of a wide selection of languages on the system necessitates the need to define communication protocol among multi-language program modules. The student and common variables of the TUTOR program can be treated as common area in a FORTRAN or PASCAL program. It is very difficult to permit core communication to other interpretive languages such as BASIC and LISP and hence, disk based communication is used.

On the system level, the desire to allow the mixture of various language modules requires the support of run time linking of programs. Commands to support this feature include "jump out" and "do out" which are smiliar to "goto" and "call" in FORTRAN except the action subroutine may be written in a different language and resides in a different binary module.

3. The Implementation of the TUTOR Language

Although our system supports multiple languages, TUTOR is optimized for CAI and thus, in this section, the TUTOR language will be discussed in detail. The description of the implementation and features of the other languages supported on our system is not within the scope of this paper.

Appendix A lists the commands which are currently available in our implementation of TUTOR. Additional commands are being implemented based on the demand of our application programs. Thus, the available commands should not be considered an exhaustive list of what can be supported but rather indicative of the kinds of commands which can be supported. Because of the large number of commands in the TUTOR language, it was decided that the implementation of TUTOR should initially have representative commands from every major command category in the language. Thus, no category of commands is fully represented in Appendix A and none has been entirely left out.

At the present time, no major obstacle is expected for the implementation of any TUTOR commands, except those which make explicit use of the existence of extended core storage on PLATO. In particular, the multi-line common statement implying a complex mapping of common variables on each time-slice will not be allowed.

There are many design constraints imposed by the desired system performance and the TUTOR language itself. It is in the context of these constraints that the current PLATO IV implementation as well as our implementation will be discussed.

3.1 Constraints imposed by desired system performance

THE TUTOR PROGRAM

The TUTOR language is most effectively utilized in a multiuser time-shared environment where the extensive user interface supported in TUTOR can be put to effective use. Both our system and the PLATO IV system attempt to support a group of users in such a way that no user need be aware of the existence of another user on the machine. One indicator of a system that is free from interference of users is the typical time required by the system in responding to even a single key. Such a response time is well under one second on PLATO IV and on our implementation as well.

The twin goals of providing fast system response time and allowing large numbers of users running concurrently on the system present design problems for the organization of central (cpu addressable) memory and bulk storage (typically block storage). Obviously, the desired performance could be maintained if all user programs fit into central memory at the same time, but this solution is not economically feasible at present. Thus the management of central memory is of critical importance to the performance of the system and the way in which bulk storage is used to supplement central memory will have an impact on the system response time.

The TUTOR language can be loosely described as a series of calls to specialized system subroutines (with arguments to the subroutines specified in the command tagfields) and simple computation. The desire to minimize central memory usage immediately requires that the library of system subroutines either be reentrant or at least be serially reusable (as is the case on PLATO) so that only one copy of the library need be kept in memory at a time.

Any further attempt to reduce the size of a TUTOR program at run time (and thus fit more of them in central memory) must involve the improvement of the object code density. The compilation of TUTOR programs into machine code would produce large program modules partially because of the need for standardized calling sequences for the system subroutine calls. Execution of TUTOR programs by the interpretation of the original source would be prohibitive because of the cost of lexical anlysis and run time symbol table mainatenance. Given that some kind of source compression is required, compilation of expressions to postfix form and interpretation by means of a virtual stack machine provide an obvious choice. The stack in such a scheme can also be used for system service routine argument passing and for user procedure (unit) linkage. Any temporary or dynamic storage required by the service routines can be allocated on this stack and the representation of the TUTOR program can be made totally reentrant. Both the local node and PLATO use the interpretative execution of an intermediate code representation of TUTOR with the exact form of the code tailored to the actual computer used. Both systems realize an additional savings in bulk storage if more than one user wishes to execute the same program by allowing a program to be shared by many different users, all executing different parts of it in a time-shared manner.

On PLATO, the programs are stored in Extended Core Storage (ECS), and when a user wishes to execute a portion of a program, logical units of that program are brought into central memory for execution. These logical units provide a virtual memory mechanism that allows full use of two million words of ECS by a central processor with a relatively small memory of its own. An alternative way of segmenting TUTOR programs has been proposed that involves link editing of unit clusters [4]. This scheme relies on the assumption that all units called from any unit are known at compile time; thus, it should be possible to build clusters of units such that inter-cluster linkage is minimized. This scheme is attractive only in environments where large segment sizes are acceptable. On the local node, the TUTOR program space is supported by demand paging. The use of virtual memory hardware reduces the overhead associated with the demand paging, and the appropriate choice of the size of a working set to use guarantees the desired response time by having a portion of each user program in central memory at all times. The demand paging apporach allows the effective use of up to the entire bulk storage space with only a small system overhead.

THE INTERMEDIATE CODE

The structure of the intermediate code for computational commands (calc, cals, etc.) is determined by the data and program storage requirements of TUTOR and the hardware available on most minicomputers. TUTOR utilizes 60 bit storage locations which may be typed as integer or floating point (e.g. n1 or v1). Floating point numbers are stored with the entire 60 bit precision, but integer arithmetic is performed over a 48 bit precision. On the local node system, storage locations are 64 bits per word with full floating point precision and with 32 bit precision for most integer operations. The use of 8-bit ASCII characters throughout the local node system implies that only 8 characters can be stored in a single word as opposed to 10 on PLATO IV. Because access to segmented variables and 64 bit logical and character operations require subroutine calls, all computational commands are compiled into a zero address stack machine code and not to machine language as is done on PLATO.

On the local node machine, an eight bit instruction syllable is reasonable, with 16 bit branch addresses, and 16 bit address fields when needed. Because of the existence of user, common, storage, and segmented variables, address fields must also contain an indication of which data space or subspace is to be addressed as well as the word size and characteristics of that space. The use of 16 bit branch addresses limits the single program module size to 65,536 bytes, comparable to the 8,000 60-bit words limit that is used on the classic PLATO system.

Most of the information about each of the many address spaces and segmented subspaces can be stored in a table of space characteristics. With this scheme, each memory address must contain an index (8 bit) into this segment table as well as the 16 bit offset into the desired addressing space. It should not be difficult to extend this scheme to multi-dimensional arrays as has recently been done on PLATO.

The assignment of values to entries in the segment table would be the responsibility of the compiler if the PLATO definition of TUTOR is retained, however it is a simple extension to allow run time redefinition of table entries. Thus this memory addressing mechanism has been extended to support dynamic allocation of local variables on the stack.

Interpretation of an intermediate code similar to that outlined above can be quite fast. The code accomplishing the action specified by any particular instruction can be prefixed to code that fetches the next byte from the instruction stream, increments the program counter, and branches through a jump table indexed with the byte fetched to the next action routine. The organization of the intermediate code outlined above implies that TUTOR commands are compiled into variable length intermediate code representations. Thus either pointers to consecutive commands must always be included in the intermediate code [5] or explicit branching must be compiled for those judging commands which utilize a run time code scan on PLATO. Our implementation of TUTOR judging includes explicit branching for judging produced at compile time and thus removes the need for command linking and run time searches.

DATA AND DATA STORAGE

STUDENT(USER) VARIABLE: Each user of a TUTOR program has a unique array of 150 60-bit words that may be used for user dependent computation. These user variables retain their values when a user changes programs, and they are saved when a user is not on the system; thus, they may be used for inter-program parameters and user related historic data so long as all programs involved agree on the use of each location. In our implementation these variables are extended to 256 64-bit words. Future plans call for the dynamic expansion and reduction of user variables beyond the basic 256 words. This will enable the replacement of the "storage" variables currently in use.

COMMON VARIABLE: The common variables provided on TUTOR are to facilitate inter-user or inter-program communication. On PLATO, common variables are stored in ECS and the name of the common used in a program must be defined at compile time; to use the common variables a mapping must be established between the ECS copy and a 1500 word central memory buffer which the program can actually access. This mapping must be explicitly defined at run time when the size of the common is larger than 1500 words. On our implementation common will be restricted to 1500 64-bit words with the name of the common defined at run time and with no mapping of variables. When the -common- command is executed and the common is not core resident, a copy of the common is moved from disk to core, cataloged as an entry in the shared region library and its core image is inserted into the user's data space. When the common is referred to a second time, the same core image is inserted into the second user's data space. The core image of the common is returned to disk when it is no longer used by any user.

STORAGE: The 150 word limit on the size of user variables poses a severe restriction on the utility of TUTOR for solving many classes of problems. Because the 150 variables retain their values between sessions for any given user, it was not considered practical to expand this limit on PLATO; instead a new data space called "storage" was introduced. Storage is statically allocated for each program, but the system must allocate it dynamically to handle transfers of control from one program to another.

Storage is allocated on PLATO in ECS, and the same mapping mechanism is used to gain access to it as is used for common variables. However, storage is always deallocated when the user leaves the system. Our current implementation imposes no limit on the size of the storage variables and these variables are addressed as common variables. As mentioned earlier, it is expected that these variables will be refered to as an extension of student variables in the future.

LOCAL VARIABLE: Another class of variables in planning is the local variables which are to be implemented on a stack and, as such, dynamic storage allocation within the unit will be made possible.

SEGMENTED VARIABLE: The TUTOR language relies on the assumption that any machine word may hold any data type. Even on a 60 bit machine, this forces great ineffeciency in the storage of such items as character strings or bit arrays. It is the intent of TUTOR that a reference to an element of a segmented array should be equivalent to a reference to a machine word in all contexts. In our implementation, segmented variables are supported with power of two in length and are run time definable. In addition, a reference to a segmented variable is allowed in all cases where a reference to a simple variable is allowed.

3.2 Constraints imposed by TUTOR commands

TERMINAL INPUT AND TERMINAL OUTPUT

As mentioned in section 2.1, an "ideal" terminal was defined in software which provides an interface to real terminals and also provides the expanded functions required by PLATO IV terminal input and output. In addition to the line editing features supported by the ideal terminal, the ability to re-open input after allowing for possible comment on the input is supported. This constitutes the most radical divergence of PLATO I/O from conventional unit record

I/O. The full capabilities of the "ideal PLATO terminal" are supported at the operating system level so that they are readily available to other system processors and user programs.

FILE I/O

The inclusion of commands for file I/O represents one of the areas in which the local node implementation of TUTOR has extended the capabilities of the PLATO IV implementation of TUTOR. On PLATO IV there are two basic forms of disk file usages available to the TUTOR programmer: data sets and student records. In the local node implementation data sets are supported by analogous commands to the ones on PLATO except that the block size is 64 64-bit words and is not variable at present. Other commands have been included (input and output) for the processing of sequential files. The input command retrieves sequential records from a character sequential file stored in the system format for text files. The "output" command allows the building of a sequential text file in much the same way that the pack command can be used to construct text in memory. In addition, the ability to open and close files at run time and the ability to have more than one file open at a time is provided on the local node system. All file structures are compatible at system level, so that they are available to all programs written in any system supported language.

COMPUTATION AND STORAGE

The facilities provided by the local node system closely parallel those of the PLATO IV system for computation. System functions such as "or" and "and" are supported as well as user defined functions. The "cls" function requires special attention when converting programs to the local node system since the word is four bits longer, but other calculational commands are straighforwardly implemented.

As mentioned earlier, the common and storage commands supported on the local node system constitute a subset of the capabilities available on the PLATO sytem. The extension of TUTOR to include local variables allocated on a stack, on the other hand, significantly increases the ease of use of the TUTOR language for complex programming tasks.

PROGRAM FLOW OF CONTROL

The flow of control in a TUTOR program can be modified in one of three basic ways: 1) unit level branching and linking (jump, goto, do, join, etc.) 2) intra-unit branching and looping (branch, doto, etc.), and 3) single statement repetition or conditional execution. For categories 1) and 2) above, the local node system provides the same commands as those provided on PLATO IV. Category 3) above is generally implemented on PLATO as a separate command which is either conditional or looping or both. The command name of a conditional command is frequently formed from its non-conditional counterpart by adding a "c". Partially to limit the proliferation of individual commands and to simplify the compilation process, our implementation provides a different approach to the problem of conditional and looping statements. We have modified the syntax of TUTOR so that all conditional commands have a unified syntax and any command can be made conditional (if meaningful) merely by using the conditional syntax in the tagfield. Thus a conditional -at- command with any mix of fine and coarse coordinates becomes possible. By enforcing a corresponding syntax for looping commands, it also becomes possible to specify the repetition of a single command as a standard part of its tagfield. Certain meaningless combinations (such as conditional looping unit) are not allowed, but all meaningful combinations are allowed.

JUDGING SEQUENCING CONTROL

The response judging capabilities in TUTOR serve two separate purposes. First, they provide the user with well structured access to a set of powerful primitives for requesting terminal input, rejecting that input, or requesting that the input be modified. Secondly, they provide access to a powerful set of character string and numeric expression analysis facilities for the evaluation of the terminal input.

The Control structures that TUTOR provides for response judging are described by the PLATO project in terms of a number of program execution states, where each command may have different meaning in each state [1]. The description of the judging control structure in terms of execution states, markers and searches for various commands obscures the underlying control structures to the extent that most beginning and many experienced TUTOR programmers never fully understand it. In this section, we will attempt to explain the structure of the TUTOR judging commands and to introduce additional rules required to simplify its structure and hence, its implementation.

THE TUTOR JUDGING BLOCK

From the description of PLATO judging, it can be deduced that a region of TUTOR code involving judging always begins with an -arrow- and is ended by a new -arrow-, -endarrow-, or -unit- command. Termination by an -arrow- or -unit- command is equivalent to termination by an -endarrow- immediately preceding the -arrow- or -unit-. Since a compiler can always generate the appropriate code for an implicit -endarrow- before -unit- and -arrow- commands if there was a previous -arrow- command, it is safe to consider judging only in terms of -arrow- -endarrow- pairs or judging blocks (excluding for the moment the problem of subroutines). Furthermore, the -arrow- is merely a prefix to a loop that begins with the first judging command after the -arrow- and ends at the -endarrow-, with termination occuring when judging state ends with an "ok" judgment. Given the above observations, an -endarrow- can be compiled as a conditional branch to the first judging command after the most recent -arrow-, where the branch will be followed if the last judgment was "no".

When subroutines and local -branch- commands are included in this description, it becomes much more complicated. There are only rare uses made on PLATO of the more pathological interactions between these language components, and in fact the addition of a few simple rules to the TUTOR language allows this simple description to be retained: first the -endarrow- for each -arrow- must be required to be in the same unit or program segment as the -arrow-. Second, all simple branches into and out of the range of a judging block must be forbidden.

CONDITIONS FOR RESPONSE EVALUATION

The judging commands in TUTOR fall into two classes: Those that are considered judging commands because they require the input to be ready before executing, and must therefore be executed in judging state, and those that actually perform some judgment, (possibly) ending the judging state with an "ok" or "no". The judging commands may be interspersed with regular commands but in judging state only the judging commands are executed; this suggests linking each judging command to the next in a sequence by branches that are conditional on judging state. When an -endarrow- is encountered in judging state, the judgment is "no", so the interpretation of all -endarrow-s as being preceded by an implicit -no- command is safe(the -no- command is defined as always ending judging with a "no" judgment).

EXECUTION OF BLOCKS CONDITIONAL ON JUDGMENT

With the addition of the branches from endarrow to the first judging command, and from each judging command to the next, it is rather easy to follow through to the end: Any judging command after the first one in a squence that is preceded by regular commands must be preceded by a conditional branch back to the location after any previous -specs- command, or to the -endarrow-. There is an exception to this rule when the previous judging command was -specs-, in which case the branch is always to the -endarrow-. These branches are conditional on the state being postjudging regular, and they accomplish the eventual transfer of control to the -endarrow- where the decision is made whether or not to loop.

JUDGING AND SUBROUTINE CALLS

One of the difficulties of compiling TUTOR is that it is not possible to specify that a unit is only to be used as a subroutine, as and extension to another unit, or as a main program. In fact, it is perfectly possible to use a single unit as all three. For the purposes of this discussion, it is sufficient to know that there are two kinds of subroutine calls in TUTOR: The first of these, the -do- command, is a regular command and is the most commonly used. The second is the -join- command; it is a somewhat confusing command because it is defined as being executed in all of the TUTOR execution states. If a unit is entered while in judging state (via -join-), the search for a judging command must continue which may be accomplished by following the unit entry with a branch on judging state to the first judging command (if any) in the unit, or to the end of the unit if none.

When judging is ended within a unit that has been attached as a soubroutine by -join-, control must somehow return to the appropriate place. In that the compiler does know how the unit is to be executed, it will be attempting to create a branch to the previous -specs- command, or some later -endarrow-. One alternative is to define -endarrow- (even if implicit) differently if it is encountered before an -arrow- in a unit or if there is no -arrow- in the unit; the compiler may then link all regular blocks (regular commands delimited by judging commands) controlled by judging commands to a possibly virtual -endarrow-, with the appropriate state marker being set before the return.

The execution of -join- while in search state, that is while searching for the (possibly implicit) -endarrow- after an "ok" judgment, is not a widely used feature of TUTOR, and it was not considered necessary to treat it here.

The problem then is to differentiate all the different execution states that may exist on return from a unit attached by -join- with conditional branches to the appropriate places: One branch should be to the next judging command if still in judging state, the other to the next -endarrow- if in post "pseudo" -endarrow- state, and finally no branch if in any of the regular states.

From the detailed discussion of various judging contingencies above, it is possible to demonstrate that all useful properties of TUTOR judging can be compiled in much the same way as a case statement. The entire judging block from -arrow- to -endarrow- consists of a repeat loop until the judgement is "ok" surrounding a case statement where the judging commands which can end judging are the case selectors, and the execution of the regular commands selected is controlled by whether or not judging was ended by the case selector. In much the same way that the entire branching structure of a conventional case statement can be defined at compile time, our implementation of TUTOR provides compile time resolution of all branches which occur in the judging process.

4. Summary

A minicomputer based CAI system with many features similar to that of the PLATO has been designed, implemented, and in operation for the past year. This system utilizes a Modcomp IV computer (Modular Computer Corporation of Florida) with 96k 16-bit words of central core memory and is supporting six terminals (three PLATO IV graphic and three ASCII alphanumeric CRT terminals). The cost of such a system is estimated at $100,000 for six terminals, $200,000 for 20 terminals, and $400,000 for 50 terminals (excluding the costs of terminals).

Several lessons developed on the PLATO system have been converted to the minicomputer system with less than 5% of the original lesson development time. Some of these lessons are from within our group, others are from the PLATO project and Purdue University School of Pharmacy.

The one hundred plus commands currently available on the minicomputer system constitute one third of all TUTOR commands on the PLATO system. However, due to the fact that these are the most commonly used commands, experience has shown that they are sufficient for most lesson development. Furthermore, additional commands can be implemented at the rate of approximately one man-week per command.

With many general purpose and CAI languages avaiable on the system, it is hoped that the author and student will have one single system that satisfy most of their needs in CAI system. The existence of multiple languages in one single system may also encourage the comparision of different CAI language features and thus accelerate the development of a standard language for CAI applications.

References

[1] Sherwood, J. A.,"The TUTOR Language," PLATO Publications, Computer-based Education Research Laboratory, University of Illinois, November, 1974.

[2] Alpert, D. and Bitzer, D. L., "Advances in Computer Based Education," SCIENCE, vol. 167, March, 1970.

[3] Chen, T. T. et al., "A Depository Health Computer Network," Advanced Medical System--The Third Century, Society of Advanced Medical Systems, November, 1976.

[4] Brackett, J., "Final Report - A TUTOR Minicomputer System," Item no. 0002AD, Contract no. MDA903-74-C-0288, ARPA Order no. 2517, Softech Inc., Waltham, Mass. February, 1975.

[5] Schuyler, J. A., "The Complete Guide to HYPRTTUTOR," Computers and Teaching, Norhtwestern University, June, 1974.

[6] Thompson, K., "QED TEXT EDITOR," Bell Telephone Laboratories, Murray Hill, New Jersey, March, 1968 (unpublished).

APPENDIX A
Alphabetical List of TUTOR Commands

add1
altfont
ansv
answer
arrow
at

back
back1
block
box
branch
bump

calc
calcc
catchup
char
clock

data
datain
dataout
data1
date
define
disable
do
dot
doto
draw

else
elseif
enable
end
endif
endarrow
entry
erase
eraseu
exact
exactc
exactv
exit

find
force

goto
group

if
imain
in
inhibit
input

jkey
join
judge
jump
jumpout

keytype

lab
lab1
loada
long

match
mode
move

nextnow
no
noword

ok
okword
openi
openo
out
output

pack
pause
plot

randp
randu
rat
ratnm
rbox
rdot
rdraw
roringin
rotate

search
setperm
showa
showe
showh
showo
showt
showz
size
slide
specs
storage
store
storen
storeu

touch
transfr

unit

write
writec
wrong
wrongv

zero

$bugoff
$bugon
$listoff
$liston
$start
$stop

1/2 OF 1/2 - GREATER THAN THE WHOLE?

Leland A. Szklennik

When people ask about the future of audiovisual and telecommunications, nine times out of ten it's the science fiction future they mean: pole-to-pole satellite transmissions, video wrist watches, wireless power transmissions, or channels that can be changed by mental telepathy. Buck Rogers and "Star Trek" have conditioned the public to dream of electronic luxuries that never seem to arrive. The communications industry is racing toward some wild and fanciful goals that would make the video-phone look like two tin cans connected by a string. One thing we are finding is that as things get smaller, they seem to become more capable.

Today's audiovisual message will be somewhat McLuhanistic since we will "view the most recent past through the rear vision mirror as we march backward into the future."

The educational system, by virtue of its size and specialized functions, has extraordinary demands for internal communications. Today, much of this communication is channeled to specific destinations by print and telephone. In the next decade, though these will continue to be the major channels of communication, the invention and development of electronic telecommunication and training devices will have an increasing impact on the entire educational structure. Lasers, fiber optics, holography, optical digital recording, and videodiscs will revolutionize the distribution, storage, and transmission of data. These techniques are now in the process of a steady, natural growth. They will expand gradually, new techniques replacing those now entrenched, as people become more familiar with the new technology and its capabilities. Let's explore a few of these new technologies.

Lasers emerged from the world of science fiction less than two decades ago and today they are making their way into the more practical world of applied technology. An important step in laser applications today may well be in the optical transmission of data. Since light has a greater capacity for transmitting information, the laser offers exciting new capabilities as an audiovisual communication medium. Solid state lasers of gallium arsenide are capable of frequencies in the gigahertz spectrum. The means of transmitting the modulated laser light is fiber optics. Fibers have enormous signal-carrying capacity for their size. One hundred fibers might be bundled into a cable no thicker than a pencil. Yet each fiber in the bundle, carrying signals as rapid bursts of light, has the capacity for many thousands or, theoretically, even millions of voice channels. By comparison, 22-gauge twisted-pair wire can carry 48 one-way voice channels. A coaxial cable might carry 5,400 one-way channels. Recently, Corning Glass Works and Bell Labs announced breakthroughs in producing laboratory optical fibers with light losses down to one or two dB/km. Such attenuation figures would mean repeaters could be spaced every 10 miles. That is better than current wire and coaxial cables, which require a repeater every few miles. This feature becomes even more important as microwave (air) routes into our cities become congested and as underground duct space for conventional cable installations becomes ever more expensive.

Fiber-optic systems will eventually make great changes in the whole field of electronics and signal processing. Before too long, for example, computers will chatter with one another over fibers; your phone calls may flow with thousands of others on wires of glass. At some point, a "wired home of the future" could give you access, over fibers, to vast amounts of information and images from libraries (in one second, a super-capacity optical signal could flash 200 books to you, letter by letter), unlimited TV channels, picture-phone calling, video shopping, and dozens of other services. Optical fibers seem destined to play an important role in the future of communications because sand, their basic ingredient, is about the cheapest thing going.

Holography was given practical impetus with the laser's invention in 1960. A hologram is the result of the pattern created by the interreaction of two light beams with each other. A more complicated use of holography is a double exposure technique called interferometry. It begins with the making of a simple hologram. Then the hologram is set up in such a way that its image falls directly on top of the real object. The resulting interferogram is a double exposure, a holographic superimposition of the 3-D image and its real precursor. This may seem a particularly useless procedure, since holography's major claim to fame is that its image is virtually the optical duplicate of the real thing. Why superimpose two identical images? Because in this way one can determine with exceptional accuracy the exact effects upon the real object of all sorts of different operations that can change it slightly. For example, by setting up an interferogram of an airplane engine part and then applying stress to the part, it is possible to see exactly how the part is affected by use, where its greatest weaknesses are, what deformations occur, and much more. Another interferometry technique, called contour mapping, maps the size and shape of objects at rest with extreme precision. The potential importance of contour mapping might best be illustrated by an example from the world of art. A few years ago, the invaluable Pieta of Michelangelo was smashed. If a holographic contour map of the masterpiece had been available, the masterpiece could have been restored not just almost as it once looked, but exactly. This technique has possibilities for the future in medicine; one of these is the making of dentures. By mapping a patient's gums, a dentist could get an exactly accurate 3-D guide upon which to pattern denture fittings. Sonic holography is another entire branch of holography that uses interferometry in a different way and achieves strangely different results. A sonic hologram shows how the inside of an object looks, in three dimensions, and in great detail with a clarity far beyond any other method. It is, in short, a sensitive, non-radioactive, 3-D x-ray.

Several companies are working on systems that ignore holography's 3-D ability in favor of another: its ability to store linear and other sorts of information in a more compact and indestructible manner than ever before possible. How compact? One mile of bookshelves into eight filing cabinets; the entire Encyclopedia Britannica onto a one-inch hologram. How indestructible? Tough enough that any small corner of a hologram retreived from a fire would probably be enough to restore all the information it contained. This is because each part of a hologram contains enough visual information to reconstruct the entire hologram with only some loss of clarity, but no loss of substance. Holographic information storage systems would work in much the same way as microfilm, except that recording and retrieval would be holographic instead of photographic. At the University of Michigan they are studying the potential production capabilities of a holographic microfilm system designed for school libraries. Their goal is the production of both a recorder and viewer. Chances for success are considered excellent. Holographic memories are being investigated for credit card checking in retail outlets, storage of non-digital information for businesses, and even fingerprint identification.

But this, of course, isn't the whole picture. One area that has already felt the touch of 3-D and will certainly change with its influence in the future, is education. Another is our personal lives: communications in our homes, personal entertainment, and means of gathering information. Holography can preserve the past and teach in the present; what, then, of the future? Three-dimensional TV, while considered by most holographers nearly inevitable, is still strictly experimental. Multiplexed cassettes may not be too far off. They could provide the core for holographic home intertainment centers. Plays, concerts, or discussions could be filmed and sold just like audio tapes. Holograms could store a 250-page manual on a credit card, or provide a U.S. map, city by city, on a video wrist watch. Holographic viewers in the schools and homes will give way to increasingly holographically oriented media. For example, tiny holo-newspapers, holo-textbooks and magazines that would bloom into 3-D on our convenient kitchen screen. Combined with fiber optics, cable and telephone technology, such units might eventually serve as multiple information, communication, entertainment, and reference connections to the world at large.

The video-disc concept has been recognized as the next obvious step in visual communications, and also as the most revolutionary development for education and the mass media since home TV via broadcast. Learning-training and information-transfer applications world-wide will see the videodisc supplementing, and then replacing, the newspaper, magazines, journals, and many other print media formats, and eventually both broadcast and cable or even satellite TV. Today, there are two basic types competing for world markets: mechanical capacitative and optical scan. The mechanical capacitative players will be less expensive, however their discs are not as durable and the system does not permit ease of browsing through the instructional materials. The optical scan players can play up to 30 minutes of video per side. All types of material browsing techniques are available.

Videodiscs are extremely durable and provide very high density storage of information. There is also the possibility of more sophisticated interactive applications through a keyboard and a digital microprocessor. Latest industry estimates indicate that a consumer could acquire a compatible playback unit for about $500, with program discs from $2 for 10 minutes up to $10 for 90 minutes. Cost factors, both for original materials and production of the videodiscs, and then for handling and transport, are dramatically lower for the disc than for any other tape-based format using magnetic materials. The videodiscs at this time will not become a new local recording medium, but a cost effective playback medium. Large disc savings will be realized from massive duplication, distribution, and utilization of the software.

Optical digital recording system is a technique for high density storage of binary information. Optical digital recording proposes to record 30 minutes of color video, in digital form, on twenty feet of 4 to 5-mil thick coated film. As much as 40 or 50 minutes of 4-channel discreet quadrophonic sound could be recorded on a three-inch square record. Reproduction could be by optical or pressing processes. Playback units can be made cost competitive with standard audio cassette recorders and a professional model should cost around $300.

Branching techniques with the combination of mass high density storage with audio and video on the same record could simplify hardware requirements for individualized instruction. Technical manuals with motion could be stored on this system. For example, 30 linear feet of maintenance manuals could be stored on a 4x4-inch piece of film. Archival materials, both still and motion, could be recorded on ceramic based materials which are water and fireproof. As an example, some 45,000 hours of motion picture film could be stored on 170,000 5x7-inch "cards". The storage space required would approximate that of a standard office desk. Through the use of 20 carrousel type machines, all material would be immediately available on a random access basis. Still photographs could be recorded on a metallic based tape and placed in tape drive units similar to those used in ADP applications. A buffer could be used for either the motion or still archives to allow the master record to be searched by others.

The videodisc technology will provide for cheap audiovisual information distribution, rapid duplication and reproduction, and simple playback through existing TV sets. Once the new videodisc format has settled into the mass media markets, you will see the slow death of conventional film and video tape hardware and software.

In the present day state of the communications art, digital technology is expanding the capacity and the reliability of wire and broadcast systems. Coaxial cable, microwave links and satellites are making it possible to transmit audio, video, and alpha-numeric data instantaneously to any point on the surface of the earth, or to make the information available to a large number of users simultaneously. The explosive growth of computer technology gives simple low-cost access to computer files, text editing, and automated data processing. Cable television has pioneered the two-way capability for video signals. An assemblage of system elements is available to meet future requirements. The market for the new information technologies is expanding rapidly, resulting in product improvements -- smaller size, increased capability, and greater reliability -- and a progressive reduction in costs; and prototype systems have already demonstrated the integration of these separate capabilities.

The educational community will be increasingly dependent on rapid communication and automated data processing. Thus, the present and future requirement is for a complete communication system including audio, video, data, and interactive capability. The "wired city" concept multiplexes communications onto a single coaxial cable. Starting as a supplement to the existing telephone plant, it can act as the backbone for a broadband, multimode, two-way interactive, highly interconnected communications system. This, in turn, is the kind of system which can be more responsive to both the human needs of the general population and the functional needs of the educational community. Many civilian communities are investigating and experimenting with this new CATV communications medium.

Training requirements are also changing. As far as delivery systems are concerned, on one hand we see increasing operational use of stand-alone equipment (VTRs and audio-synchronized slide and filmstrip projectors), while at the same time increasing efforts are being expended on remotely accessible resources. Thus, the current trend toward stand-alone training aids may well reverse itself toward remote telecommunications-based systems, particularly as two-way CATV attains the individual responsive characteristics of a stand-alone system. Although these delivery systems are being used with more and more success, some significant problems are yet to be confronted. Among these are maintaining up-to-date products (films and video tapes), proper design and use of the instructional material, and the large investment necessary to produce a wide variety of courseware and software. In addition, the effective utilization of individually paced instructional techniques will depend on a positive attitude toward its use by both the students and the instructors.

Advances in educational technology, the systems-engineering approach to instruction, self-paced or individualized instruction, and the use of computers in teaching suggest that application of such instructional methods and media capability could significantly increase the flexibility and effectiveness of teaching and reduce costs as well. Today, in wading through the proliferation of obselete equipment and the multiplicity of current devices, the requirement for a single device is increasingly evident. A color television receiver and typewriter keyboard could provide the necessary display and interactive capability for accessing audiovisual materials, computer-assisted instruction, automated data processing, or videodisc. There is no question, however, that if technology problems can be overcome, the ability of the educational systems to teach efficiently will be greatly enhanced. The question is no longer whether it can be done, but when and how it will be done. Perhaps the most significant technological breakthrough within the next decade for the field of education will be the utilization of these new technologies. I would like to echo Dr. V.K. Zworykin, who is widely recognized as the father of television, in his challenge to educators concerned with the utilization of internal communications: Match the ingenuity and innovation of the engineers and scientists who develop television's technology; don't be afraid to dream about going beyond the confines of the known, the conventional, and the convenient in your use of audiovisual media for education.

Instructional television is not competitive because of its limited budgets and audiences. Its expected impact on American education has been largely negated by poor acceptance by the school community as a whole - students, teachers, and administrators. Students object to crude and unimaginative programs produced with shoe-string budgets. Their experience with commercial television establishes their expectancy for interest level and production quality. Teachers object to synchronizing their schedules with TV, and often have disagreements with the content and methodology. Administrators often do not understand either the potential or the logistics of televised instruction, and fail to provide the appropriate level of support. And so ITV, where it exists, tends to be inadequate. But, for all the investments of time, effort, and money, for all the equipment available, for all the research and evaluation results (so many of which report "no statistically significant difference"), for all the experience at all the educational levels, ITV is unfortunately still too sparingly employed. Despite the many problems which have these past two decades plagued ITV, I still prefer to remain optimistic.

Until we recognize and begin to come to grips with the vested ego-interests inherent in traditional methods of teaching, we cannot realistically deal with the problems which surround the introduction of media into formal education. "Today" is 30 years into television, more than 50 years into radio, and 75 to 100 years into film and audio recording. Implementing these kinds of Buck Rogers, "Star-Trek" systems will not be a matter of simply waiting for the hardware, but revolves around a still infant instructional technology.

There is the tendency, of course, to get carried away with all the possibilities. It could be 20 or 30 years before a majority of American families have "entertainment centers", if they ever do. It took 17 years for a majority of American households to acquire color televisions once they were put on the market, on the other hand, it took only ten days for eight million people to rush out and see "Jaws".

Mr. Leland A. Szklennik
DoD - DAVA
1117 North 19th Street
Arlington, Virginia 22209
(703) 694-4611

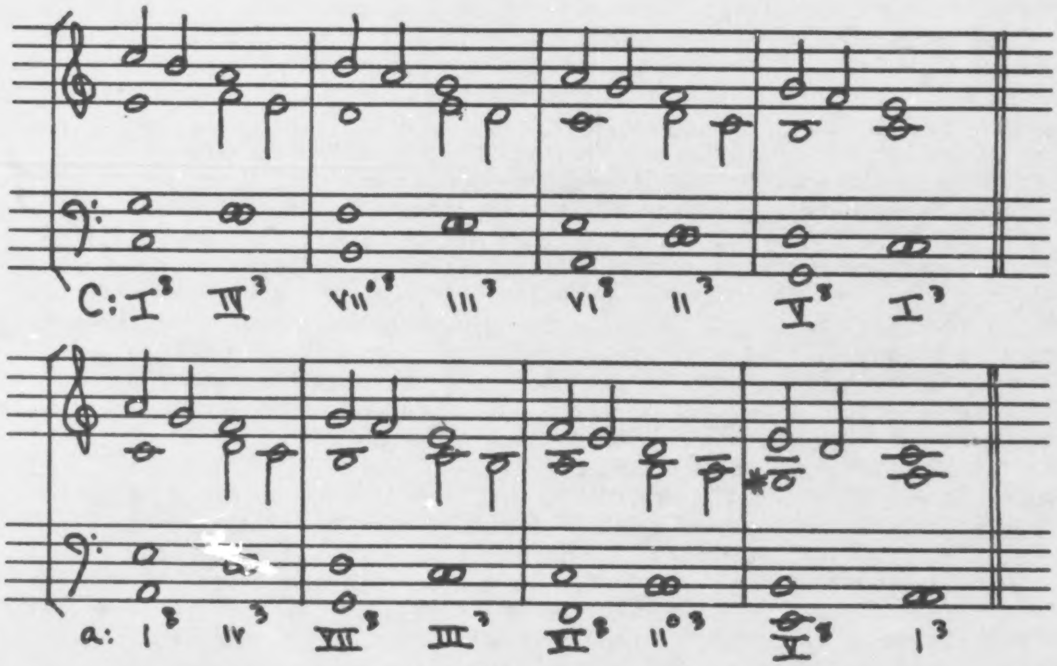
TWO IMPORTANT PRINCIPLES FOR CONSTRUCTING COMPUTER-BASED MUSIC INSTRUCTION PROGRAMS -- THE CIRCLE OF FIFTHS, AND KEY TRANSPOSITION

by

Marvin S. Thostenson

Among the first percepts experienced and learned by students studying tonal harmony are those of the tonic chord, the dominant chord, and the resolution of the dominant chord to the tonic chord. This literally "dominating" progression is the mainstay of the authentic cadence, V I, and of the various forms of the full cadence. Through further application of its root movement principle of a downward fifth, the chord series becomes the extended basic progression of common-practice harmony (1). The familiar sound of Example #1 gives a simple illustration of the basic progression extended to cover all the chords of the diatonic scale (2).

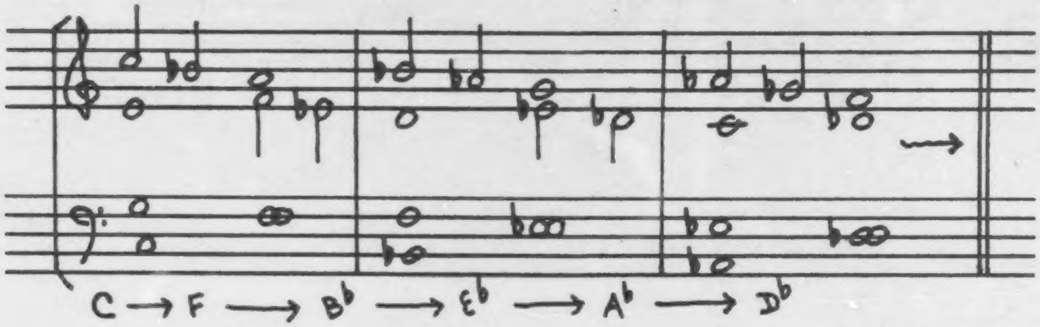
EXAMPLE #1


C: I8 IV3 vii°8 iii3 vi8 ii3 V8 I3
a: i8 iv3 VII8 III3 VI8 ii°3 V8 i3

It is well known that series of two, three, or four and often more successive harmonies from tonal music's extended basic progression series constitute from about 40% to as much as 65% of all the chord progressions found in a given composition from the common-practice period (3).

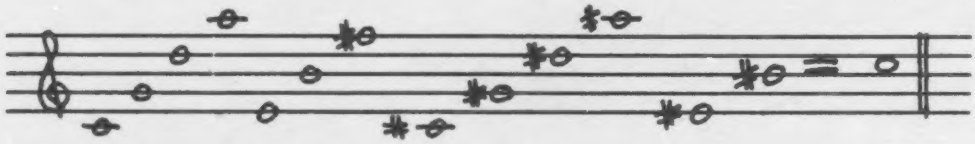
Just as the foregoing basic progressions remain centered in a given tonality when used diatonically, the series becomes a succession of modulations when only downward perfect fifth and fourth root movements are used. The resulting group of keys, a major second lower at each bar in Example #2, is an abbreviated illustration of the harmony found in occasional modulating sequences.

EXAMPLE #2



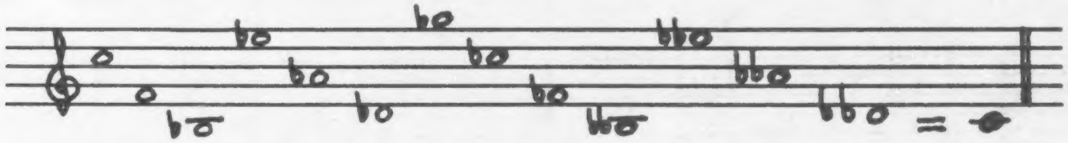
The importance of the perfect fifth was already known to Pythagorus in his acoustical studies done about 550 B. C., during which he invented the Pythagorean scale based on pure fifths (4). A spiral of 12 upward pure fifths closes on B#, which ignoring octaves is a Pythagorean comma, or 24 cents, about one-fourth of a half-step higher than C. In essence, tempered tuning removes the Pythagorean comma by trimming two cents from each pure fifth to yield a chromatic scale based on a circle of 12 tempered fifths. See Example #3.

EXAMPLE #3



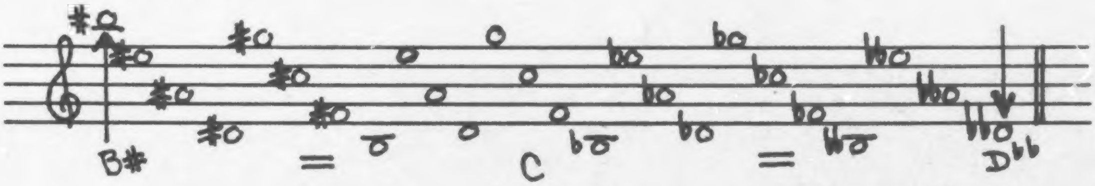
The chromatic scale formed from the circle of fifths has but one accurate interval — the perfect octave. However, its adaptability to all keys has made its use practically universal. Writing a similar circle of perfect fifths downward from C leads to a Dbb that is also enharmonically equivalent to C. This set provides another group of pitch names for the notes of the chromatic scale. Example #4 follows.

EXAMPLE #4



These two series of perfect fifths combine to provide a string of 24 perfect fifths from c's upper enharmonic, B#, to C's lower enharmonic, Dbb. See Example #5.

EXAMPLE #5

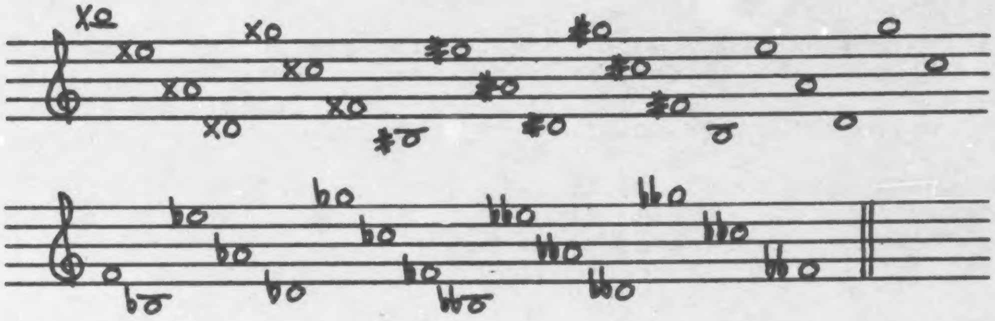


It will be noticed that no pitch name is repeated in this series, but it does include a number of the enharmonically equivalent note names for the 12 chromatic pitches. Extending this series of fifths to include all double-sharped and double-flatted note names makes available all possible names for the pitches of the chromatic scale. Further extension upward would introduce F-triple-sharp, and downward, B-triple-flat, both obviously impractical spellings. See Example #6, which is written both in letter names and music notation.

EXAMPLE #6

Bx Ex Ax Dx Gx Cx Fx B# E# A# D# G# C# F# B E A D G C F

Bb Eb Ab Db Gb Cb Fb Bbb Ebb Add Dbb Gbb Cbb Fbb



The availability of all possible spellings for the chromatic scale's 12 pitches, each one unique, is the compelling reason for utilizing the series of downward perfect fifths as a source string for writing computer programs. Example #7 summarizes the variant spellings for the 12 pitches of the tempered scale.

EXAMPLE #7

C	C#	D	D#	E	F	F#	G	G#	A	A#	B	C
B#	Bx	Cx	Eb	Dx	E#	Ex	Fx	Ab	Gx	Bb	Ax	B#
Dbb	Db	Ebb	Fbb	Fb	Gbb	Gb	Abb	*	Bbb	Cbb	Cb	Dbb

* Ab/G# is the center of the diatonic tritone, F to B. Since another enharmonic equivalent based on either F or B would involve either a triple-flat or a triple-sharp, there can only be two spellings for this pitch, namely, Ab-G#.

The second reason for using the circle is related to the first, namely, that each musical structure, for example a triad, can be described in terms of its own unique pattern of relationships on the circle of fifths. For a simple illustration, look at the short series of perfect fifths shown in Example #8. Each note, written in letter names for programing, is two characters wide, that is, each is comprised of a letter followed by the appropriate accidental sign. In Example #8 spacing is added for visibility.

EXAMPLE #8

G#	C#	F#	B	E	A	D	G	C	F	Bb	Eb	Ab	Db	Gb	Cb	Fb
-16	-14	-12	-10	-8	-6	-4	-2	0	+2	+4	+6	+8	+10	+12	+14	+16

Take note of the unique pattern that spells the C major triad. Find C, move to E, four pitches to the left, and then move back three pitches towards C to G. This same pattern, begun on any pitch of the series, produces the major triad spelled on that pitch. Using the extended series of Example #6 makes it feasible to spell all possible major triads.

In computer programing the relationships are often given in numerical form. Working from any point as a zero, numbers to the left are considered to be negative and those to the right positive. Thus, converting the unique major triad pattern on C gives the numbers 0, -8, -2, as found in Example #8. The patterns for the four triad types are listed following:

- major triad ----- 0 -8 -2
- minor triad ----- 0 +6 -2
- dim'd triad ----- 0 +6 +12
- aug'd triad ----- 0 -8 -16

These positive and negative constants from C can readily be programed to provide the desired triad quality at any pitch on the circle of fifths. See examples, last paragraph, page 6.

It can now be deduced that each pitch concept — intervals, scales, and all possible triadic as well as other chordal structures will each have its unique pattern of pitch relationships relative to some given starting point. Since C is so central to musical thinking, it is used as the logical starting point. Any other pitch as a starting point is, as has been mentioned, different from C by a negative or positive constant that is one form of the transposition factor. This point is easily reinforced if we notice that B is -10 from C (G D A E B, going up by negative 2's from C), and that Ab is +8 (F Bb Eb Ab, going down by positive 2's from C).

Using Example #6 as the source string, take note of how intervals relate. Beginning the fifths down positively from C, we have the following: (Measure each interval from C.)

	C	F	Bb	Eb	Ab	Db	Gb	Cb	Fb	Bbb	Ebb	Abb	Dbb
-up	P8	P4	m7	m3	m6	m2	d5	d8	d4	d7	d3	d6	d2
-dn	P8	P5	M2	M6	M3	M7	A4	A1	A5	A2	A6	A3	A7
	0	+2	+4	+6	+8	+10	+12	+14	+16	+18	+20	+22	+24

Examining this set brings out (1) that each vertical pair of upward and downward intervals is complementary, and (2) many, but not all possible interval names are found. Next, consider a similar series of 12 fifths approaching C from the left, that is, from the negative side. (Measure from C!)

	B#	E#	A#	D#	G#	C#	F#	B	E	A	D	G	C
-up	A7	A3	A6	A2	A5	A1	A4	M7	M3	M6	M2	P5	P8
-dn	d2	d6	d3	d7	d4	d8	d5	m2	m6	m3	m7	P4	P1
	-24	-22	-20	-18	-16	-14	-12	-10	-8	-6	-4	-2	0

Now dealing with both series of intervals together, (1) all up and down interval pairs are complementary, (2) each pair of one set are inverted when compared with the corresponding pair of the other set, and (3) considering upward intervals or downward intervals as separate sets, the symmetrical positive and negative forms are the same numerical size and complementary in symbols. The hand out, Spelling Intervals, Upward and Downward, provides a simple means for demonstrating these points visually. Note on the template for upward intervals that every vertical pair is complementary. It can also be seen that vertical pairs equidistant from the central solid arrow are symmetrical and inverted. For computer programing, each successive symbol to the right adds +2 and to the left adds a -2. Hence, the 4th symbol to the left, a -8, is an upward major third, whereas, a +8 upwards is an upward minor sixth.

The second hand out, Triads and Chords, is similar to the first except that percepts other than intervals from music fundamentals and harmony are made available for study. Along the outer edges of the template a number of formats sampling various areas of music theory are found. Proper placement of these along the source string needed for both templates correctly spells each percept.

It is hardly an exaggeration to suggest that the functional completeness and flexibility, as well as the clear cut symmetry found in the circle of perfect fifths point out a marvelous consistency and even a kind of beauty, which is more evident each time we see or hear a resolution of a downward fifth harmony.

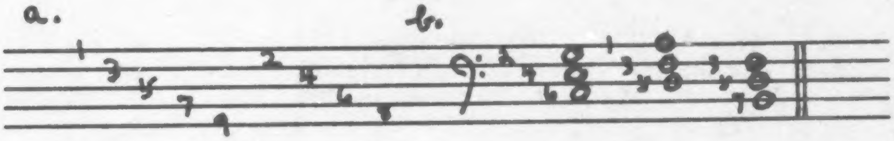
A third reason for using the downward series of perfect fifths hinges upon the transposition function. For certain operations this capability for transposition is innate within the source string of fifths. This is shown by applying a C-related pattern to other beginning pitches. As hinted above, this is accomplished in programing by adding the positive or negative factor needed to transpose the C-starting point to the one wanted. The note B is at a position of -10 relative to C, and Ab is at the relative position of +8. Two examples of triad transposition are thus shown by building a major triad on B by applying the pattern, 0 -8 -2, at -10. To transpose the same triad pattern to Ab the pattern, 0 -8 -2, is applied beginning at +8. The resulting triads are of course B D# F# from B and Ab C Eb beginning on Ab.

Thus the circle of fifths source string has been shown to be potentially capable of producing all possible pitch percepts in letter name notation, including their transposition to various keys. The print out obtained is in letter-names; print out in music notation is also possible through additional steps that are not covered in this paper. Discussion will next center on another principle that does involve print out in music notation.

The second important programing principle concerns a simple means for transposing illustrations of harmonic progressions. By this means whole note examples of from two to six chords in length, representative of the various areas of harmony can be made available in all keys. The principle that supports this method of transposition is inherent in the operation of a typewriter, or computer, both of which print out materials line after line down the page.

Considering this point, think of a music staff's five lines and four spaces as comprising nine printing positions — first print the top staff line, second, print a blank line, third, print the next line down, and so on through the lines and spaces. See Example #9a. Take note that the odd printing positions are specific staff lines — the fifth printing position gives the third line which is D in bass clef, for example. The even printing positions represent the places where notes belonging in spaces are printed — A on the first space of bass clef is the eighth printing position.

EXAMPLE #9



Next examine the b-part of Example #9, where the C major, D minor and B diminished triads are seen on the bass clef staff. The C major triad includes the sixth, fourth, and second printing positions. Note further that the D minor triad's printing positions are the fifth, the third, and the first. A principle is evolving! Subtracting one from each position of the C chord changes it -- it is TRANSPOSED to a D chord! Or, adding one to each position of the C chord converts it into a B chord with the positions 7, 5, and 3.

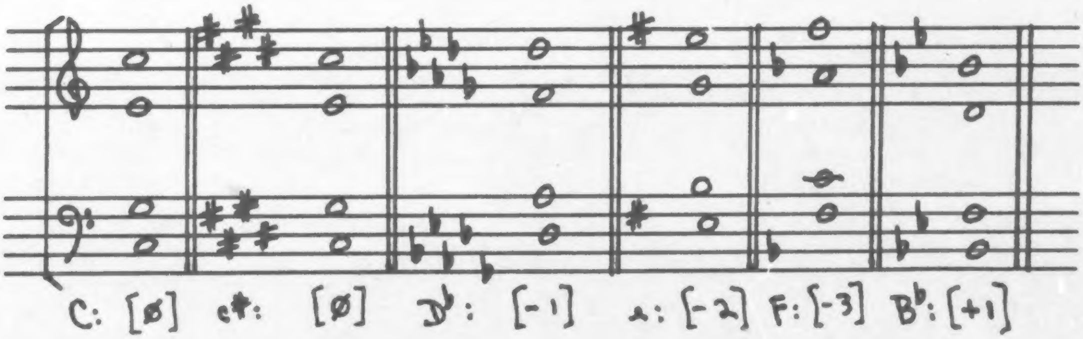
Hence, to transpose any structure from a C-orientation —

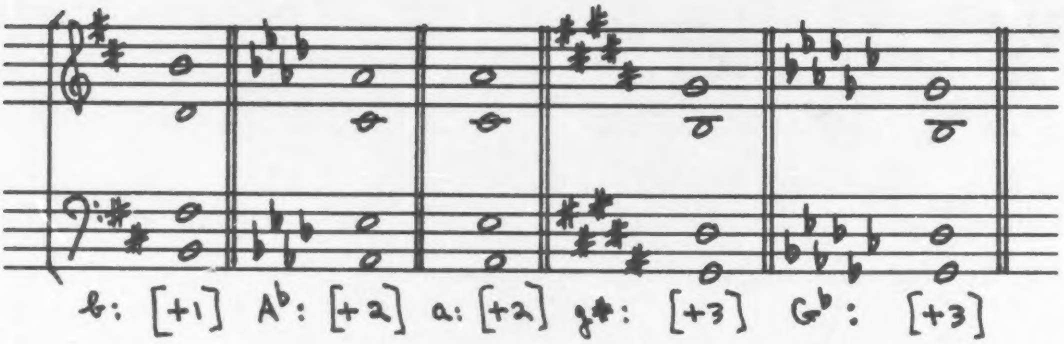
A. Add or subtract as pertinent the needed transposition factor from each print out position of the example. Observe that the six following statements account for all pitch levels.
    (1) Subtract one to change to a D orientation.
    (2) Subtract two to change to E orientation.
    (3) Subtract three to change to F orientation.
    (4) Add one to change to B orientation.
    (5) Add two to change to A orientation.
    (6) Add three to change to G orientation.

B. Affix the appropriate key signature to the transposed version. It is interesting to see that C can have no signature, one of 7 sharps, 7 flats, 4 sharps, and 3 flats, and that D can have signatures of 2 sharps, 5 flats, 6 sharps, and 1 flat, and so on for the other five letter names of music notation.

Example #10 shows a tonic chord in C (major) orientation. Successive illustrations show its transposition to a number of major and minor keys. The transposition is effected (computer-wise) by adding the positive or negative transposition factor shown below the example. This is supplemented by affixing the key signature appropriate for the desired transposition. The process for transposing longer phrases is essentially identical.

EXAMPLE #10





The above mentioned techniques, expanded, combined, and varied as needed are the basis for the computer assisted instruction programs that my assistant and I have written thus far. It has not yet been possible to add the dimension of sound, therefore our present programs deal only with the problems of music theory study. However, there is no doubt that many portions of our present programs will easily translate into usable aural skills programs. Due to the current fiscal situation it also is not yet possible to make broad use of all our available programs, but here as well we have every hope of expanding student participation in the near future. In the meantime, we shall continue as intensively and extensively as possible to develop more programs that can be useful for computer assisted instruction.

School of Music
University of Iowa

REFERENCES

1. Christ, DeLone, et al. Materials and Structures of Music, II. Englewood Cliffs, New Jersey: Prentice Hall, Inc., 1967, page 361
   Kraft, Leo. Gradus, Book I. New York: W. W. Norton and Co., Inc., 1976, pages 80-81

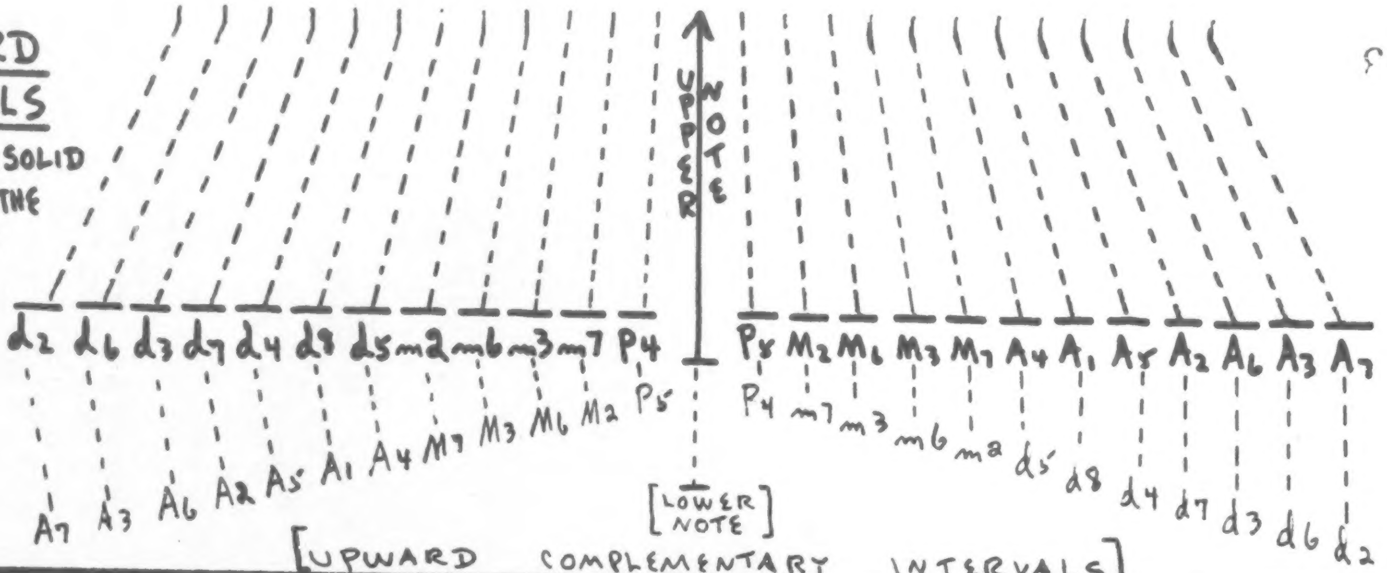
2. Siegmeister, Elie. Harmony and Melody, Volume I. Belmont, California: Wadsworth Publishing Co., 1965, pages 227, 427

3. Personal unpublished research has shown that compositions from the Baroque through the Romantic Periods are strongly based on falling fifth harmonies. Two especially rich examples are (a) Bach, "Prelude and Fugue in D Major", Volume I, Well Tempered Clavier. (b) Chopin, Etude No. 1 in C Major, Op. 10, No. 1.

4. Apel, Willi. Harvard Dictionary of Music. Cambridge, Massachussetts: Harvard University Press, 1947. "Circle Fifths", page 150; "Pythagorean Scale", page 618; "Temperament", pages 734-736.

DOWNWARD INTERVALS

-- POINT THE SOLID ARROW AT THE UPPER NOTE AND READ DESIRED INTERVAL ALONG THE DOTTED LINE.


UPPER NOTE
d2 d6 d3 d7 d4 d8 d5 m2 m6 m3 m7 P4
P5 M2 M6 M3 M7 A4 A1 A5 A2 A6 A3 A7
A7 A3 A6 A2 A5 A1 A4 M7 M3 M6 M2 P5
P4 m7 m3 m6 m2 d5 d8 d4 d7 d3 d6 d2
[LOWER NOTE]
[UPWARD COMPLEMENTARY INTERVALS]

Hand Out #1

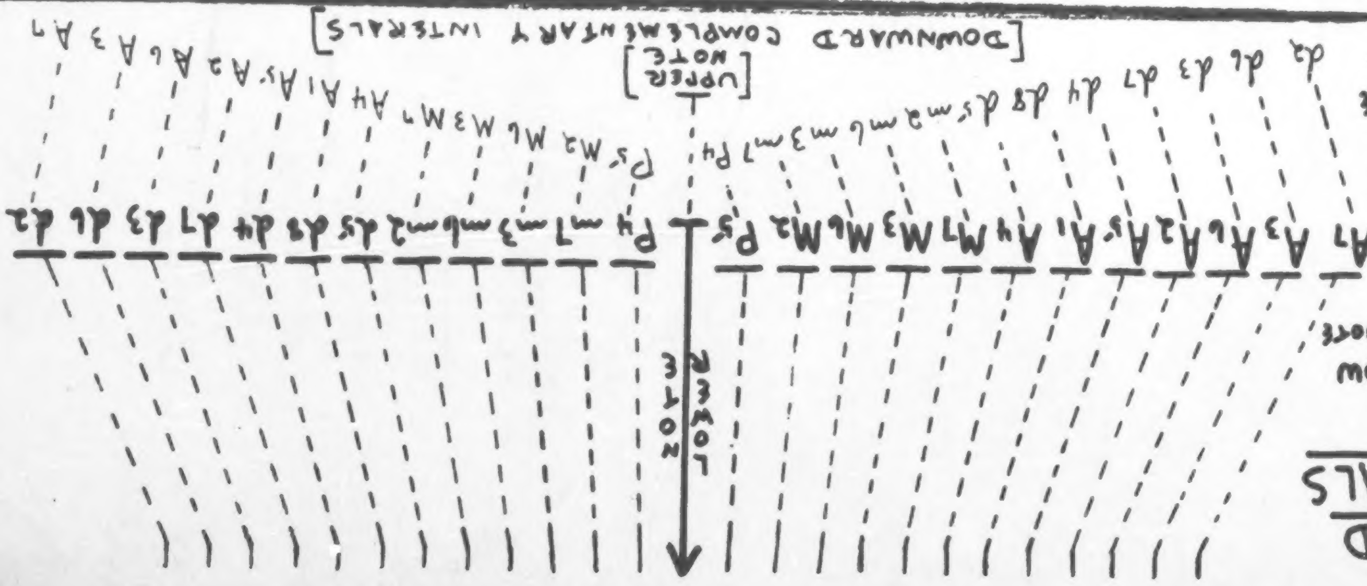
SPELLING INTERVALS -- Up, Down

by M. S. Thostenson, SCHOOL OF MUSIC,
UNIV. OF IOWA, IOWA CITY, IOWA 52242

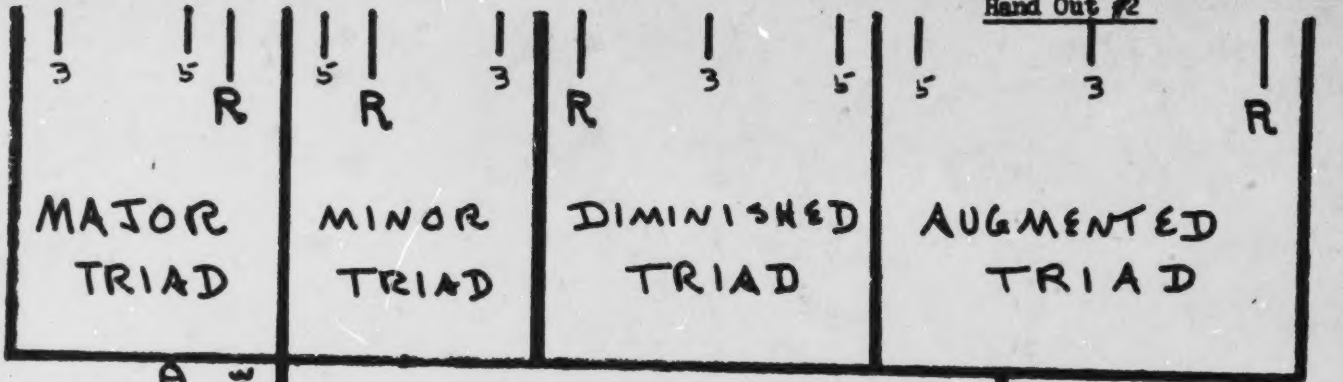
COPYRIGHT
1976

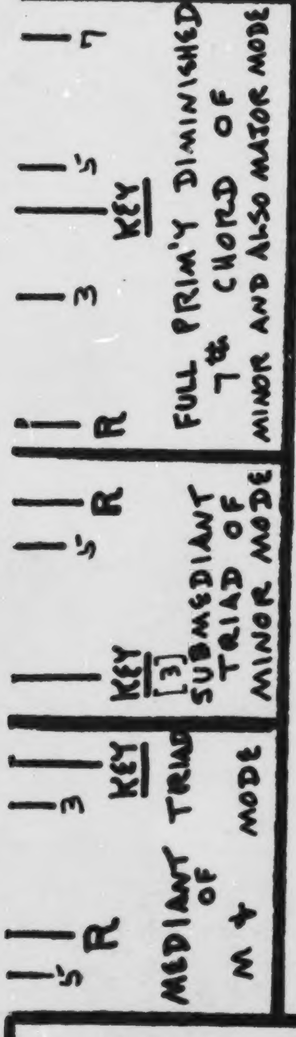
UPWARD INTERVALS

- POINT THE SOLID ARROW AT LOWER NOTE AND READ ANSWER WHERE DOTTED LINE OF INTERVAL MEETS CIRCLE OF FIFTHS.


LOWER NOTE
A7 A3 A6 A2 A5 A1 A4 M7 M3 M6 M2 P5
P4 m7 m3 m6 m2 d5 d8 d4 d7 d3 d6 d2
d2 d6 d3 d7 d4 d8 d5 m2 m6 m3 m7 P4
P5 M2 M6 M3 M7 A4 A1 A5 A2 A6 A3 A7
[UPPER NOTE]
[DOWNWARD COMPLEMENTARY INTERALS]

Hand Out #2


3
5
R
MAJOR TRIAD
5
R
3
MINOR TRIAD
R
3
5
DIMINISHED TRIAD
5
3
R
AUGMENTED TRIAD


7
5
3
R
KEY
FULL PRIM'Y DIMINISHED 7th CHORD OF MINOR AND ALSO MAJOR MODE
R
5
KEY
[3]
SUBMEDIANT TRIAD OF MINOR MODE
3
R
5
KEY
MEDIANT TRIAD OF M + m MODE

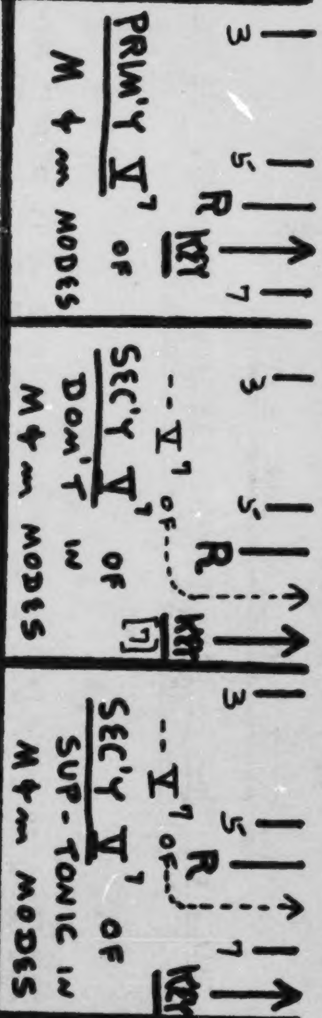
TRIADS AND CHORDS

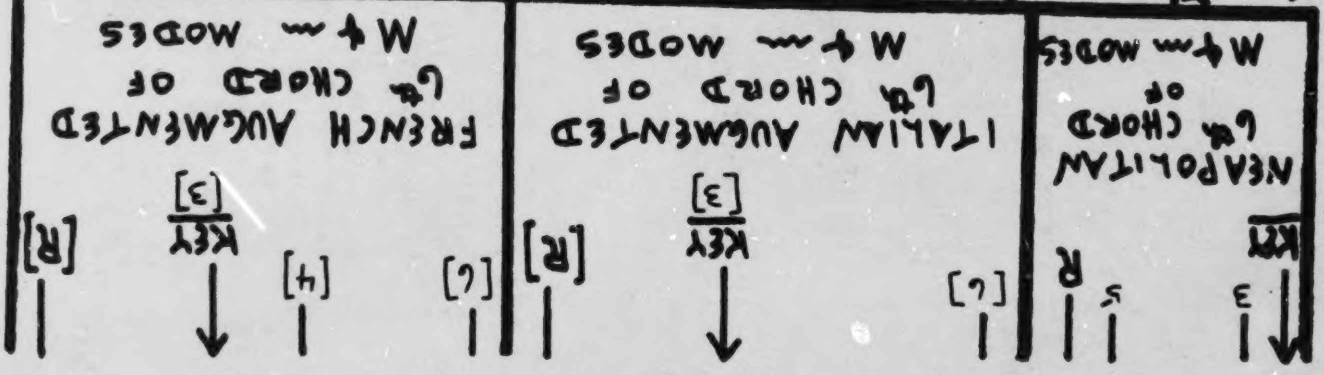
-- 13 TEMPLATE PATTERNS
FOR
FUNDAMENTALS AND
HARMONY

by

M. S. Thostenson
SCHOOL OF MUSIC
UNIV. OF IOWA
IOWA CITY, IOWA
52242

COPYRIGHT 1976


3
5
R
7
KEY
PRIM'Y V7 OF M + m MODES
3
5
R
[7]
KEY
SEC'Y V7 -- V7 OF DOM'T IN M + m MODES
3
5
R
7
KEY
SEC'Y V7 -- V7 OF SUP-TONIC IN M + m MODES


NEAPOLITAN 6th CHORD OF M + m MODES
KEY
3
5
R
ITALIAN AUGMENTED 6th CHORD OF M + m MODES
[7]
KEY
[3]
[R]
FRENCH AUGMENTED 6th CHORD OF M + m MODES
[7]
[4]
KEY
[3]
[R]

(for) Hand Outs #1 and #2

THE EXTENDED CIRCLE OF PERFECT FIFTHS

For use with TRIADS AND CHORDS -- Thirteen Template Patterns for Fundamentals and Harmony, and SPELLING INTERVALS, Upward and Downward.

Bx Ex Ax Dx Gx Cx Fx B# E# A# D# G# C# F# B E A D G C F Bb Eb Ab Db Gb Cb Fb BbbEbbAddDddGbbCbbFbb

- For Intervals -- Pointing the arrow of the template toward the desired lower or upper note, permits the other note of the interval to be read by following the line from the interval name to the note on the circle of fifths. More than 1,000 possible intervals.

- For Triads and Chords - For triads point the template arrow at the desired triad root and read the other notes on the circle of fifths.

- For functional chords the template arrow **must** point at the desired key. The root and other members of the chord can then be read on the circle of fifths. It must be understood that each different type of chordal function requires another template pattern, since each pattern is unique. *

* The following are included:

1. major triad
2. minor triad
3. diminished triad
4. augmented triad
5. primary dominant 7th chord
6. dom't 7th of the dom't
7. dom't 7th of the supertonic
8. med't triad of major mode
9. submed't triad of minor mode
10. full form of the prim'y dim'd 7th chord of both modes
11. Neapolitan 6th of both modes
12. Italian aug'd 6th of both modes
13. French aug'd 6th of both modes

by Marvin S. Thostenson
School of Music
University of Iowa
Iowa City, Iowa 52242
Copyright 1976

THE COMPLETE COMPUTER-BASED MUSIC SYSTEM:
A Teaching System - A Musician's Tool

Dr. G. David Peters
School of Music, University of Illinois

Consider this computer-based music system for the music educator and musician as a given. The current state of the "art" of technology necessary to produce this model does exist, placing this system somewhere between the possible and probable category. Undoubtedly, in the coming years many of the technological developments will make the construction of the basic system much easier, just as printed circuits, integrated circuits and microprocessors have pushed the miniaturization of electronics forward in several recent steps.

Such is the state of a rapidly developing computer technology that, what was a monumental task only five years ago, has been diminished to a long week-end project for an electrical engineering student. Granted that the original idea still takes time to develop and refine, but the execution or physical development process is a much easier road than it was only a few years ago. With these thoughts in mind, the technically naive can easily follow the "plan" for the "complete computer-based music (CBM) system" and perhaps relate the "system" to their own teaching specialty or area of music. The technically sophisticated computer scientist may find some value in the "system" in reviewing the computer needs of the musician and the music educator.

THE FUNCTION:

The complete computer-based music system has two applications: 1) to serve the practicing musician and 2) to instruct the student musician. At some point these purposes blur, in that the active musician must constantly practice skills and review materials that the student musician is only just learning. The difference in the two applications may be only in level of user accomplishment. Consider a computer-based device that could serve the practicing musician in the following manner:

for the composer being able to:

> compose and/or perform in "real'time" electronic media
>
> compose traditional music at a computer terminal, hearing the composition at will with proper timbres for each part
>
> print the full score of camera-ready quality with computer-graphics capabilities in a matter of minutes
>
> extract and print individual parts within minutes from the full score.

for the performer being able to:

rehearse music under the "supervision" of the computer which can detect wrong pitches or note durations.

practice performance of intonation or interval accuracy by either singing or playing an instrument.

practice a keyboard exercise with immediate error feedback from the computer.

see and compare a personal performance with a model in music notation.

find proper solutions for fingering problems with other technical information related to the performance of a composition.

review performance practices by period under computer supervision.

The advantage to the musician in using a computer as a tool is that of any time-saving device. The hours spent in copying music have long slowed the composer's productivity. The high expense of copying parts has precluded the performance of major compositions of many composers. The particular advantage to the composer is two-fold in that he can hear and see his music develop with much greater ease.

The performer, when approaching virtuosity is much more analytical about performance practice than any student. Error detection during practice periods is of value to any performing musician, beginner or professional. The degree of accuracy required in basic computer judgment of pitch and rhythmic performance can be adjusted, from a tolerance suitable for the first-year student to a preciseness desired by a professional performer to attain. For the performing musician, the computer-based system can allow the same visual depiction of his performance the composer finds of value, while allowing for computer drill and practice supervision.

The second, if not largest, area of application of a computer-based music system is that of computer-assisted instruction (CAI). The instruction of students in various areas of music is a logical application for the powerful computer. The very type of drill and practice needed to be a successful performer or composer lends itself to CAI. The individualized nature of CAI is the major advantage in this application. While only the computer can accomplish some of the electronic music controls for the composer, teaching is still very much a person to person, or person to group experience. The added feature of using CAI in music instruction is not meant to replace the high school band director or the elementary school general music teacher. However, the immediate feedback available from the computer is an improvement over any existing classroom situation in improving eartraining skills.

The application of CAI to music can be considered for the music student, the audience and/or the consumer of music. Successful applications of CAI to music to date include the teaching of music theory, music methods, eartraining, notation, music terminology, acoustics, jazz history, music gaming, instrumental pedagogy, music composition, and music research. The application of CAI to the music curriculum, for the most part is centered at the university campus. CAI music instruction has been utilized at both the elementary school and high school level; however, most research and development in music curriculum remains at the college level. Consider the application of a CBM system in CAI in music in the traditional areas of a music curriculum:

in music theory;

drill and practice in rhythmic, melodic and harmonic eartraining, part-writing, and keyboard skills.

instructional lessons in scale construction, mode construction, chord construction and resolution, figured bass, harmonic analysis, key modulation, structural analysis, contrapuntal analysis, and music acoustics.

in music composition;

drill and practice in harmonization, instrumentation and theme development.

instructional lessons in counterpoint, motive development, graphic notation, twelve-tone and matrix composition, electronic music, instrumentation, arranging, and transcription.

in music history;

drill and practice lessons in theme recognition, style and period identification, critical listening of performance and music terminology.

instructional lessons in music appreciation and music history related to composer, country, period, artistic trend and movements as related to social, economic, and historic development of civilization, plus instruction in listening and critiquing music and music performance and valuing music.

in music methods:

drill and practice lessons in critical incidence in the classroom, error detection in music performance, and music activity direction.

instructional lessons in classroom management, music pedagogy, learning theory, problem solving, student testing and motivation, and program administration.

in music performance;

drill and practice lessons in music terminology, music performance of exercises, skill development, error detection, and performance modeling.

instructional lessons in performance practicing, instrument care and maintenance, vocal diction and phonics, performance practices by period, and improvisation techniques.

in music research;

drill and practice lessons in test applications and test situation problem solving.

instructional lessons in test construction, scoring and rating of music performance, musicality, course, student and teacher evaluation; plus standard test and measurement procedures and standardized tests available in music.

Drill and practice type lessons are well-suited to music in many areas of instruction. The use of the "gaming" mode of instructional lessons can be used successfully with a number of younger children in the areas of music fundamentals, both in music theory and music appreciation. The instructional lessons appear in several other variations, including tutorial lessons and laboratory simulation, or inquiry lessons. Tutorial lessons present instructional questions and available tutor-type information much as an individual teacher might in working with one student. The laboratory simulation mode of instruction essentially designs a laboratory in which a student may conduct experiments and assess the experimental results. Available information is constantly under computer control. This inquiry mode of instruction is significant in the development of critical thinking in presenting the student with a problem that cannot be answered with a simple or multiple-choice response. It may call for an entire sequence of analysis and contructed responses that cannot be anticipated.

The instructional areas in music requiring the recall of cognitive materials and the acquiring of psychomotor skills are well suited to CAI. Several successful CAI centers have demonstrated the application of CAI to the teaching of music.

THE HARDWARE:

The delivery system needed to realize the two applications of the CBM system requires the combination of available components. The development of the computer to its current efficiency, operating in real-time while time-sharing, was a prerequisite. The hardware needed to present music media on any "standard" CBM system must include:

1. graphic capabilities for music notation
2. random-access playback-record unit with high-quality audio
3. digital to analog equipment for instant composition of melodic or harmonic sound generation
4. analog to digital equipment to analyze real-time music performances

In the combining of these assorted parts into an integrated system some consideration of the instructional and musical use of such a system must precede the construction. In following the construction from a general system to a music system the given components are a computer and a terminal (figure 1):

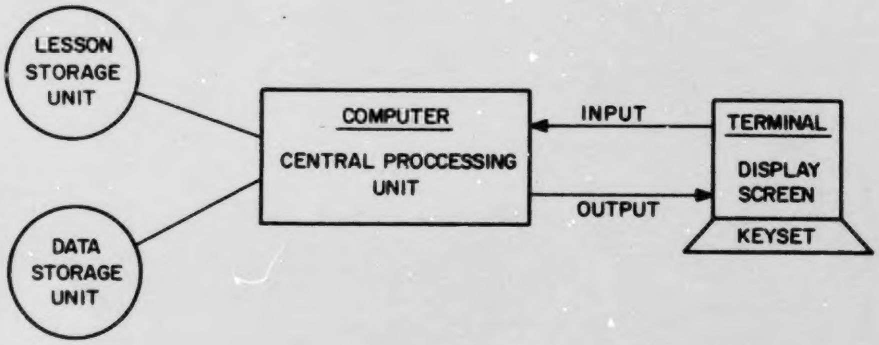
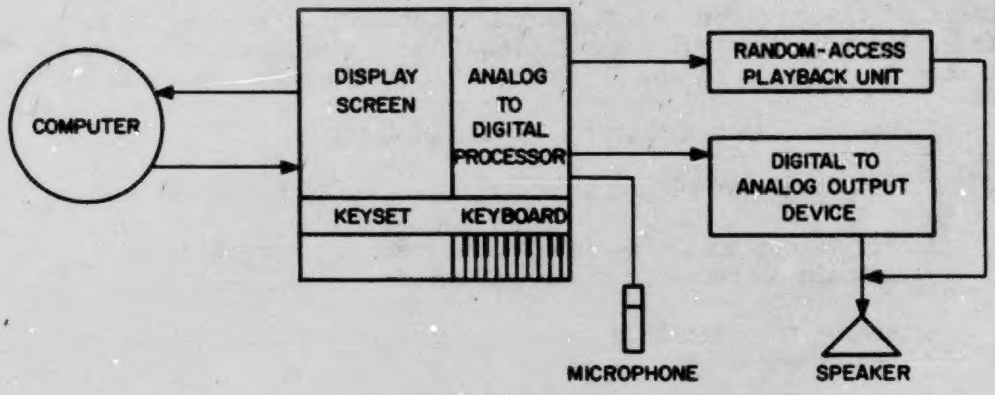


Figure 1

Regardless of the size of the computer used with the CBM system, questions of adapting a working computer system to music instruction and performance yield the following options:

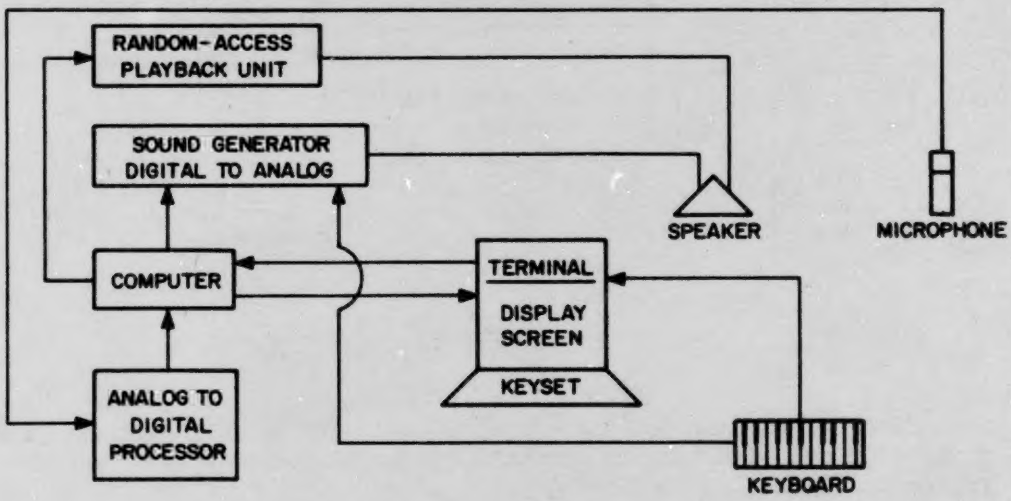
1. the complex, "smart" terminal which performs many operations before the refined and analyzed information is sent to the computer for treatment (figure 2).


COMPUTER
DISPLAY SCREEN
ANALOG TO DIGITAL PROCESSOR
KEYSET
KEYBOARD
RANDOM-ACCESS PLAYBACK UNIT
DIGITAL TO ANALOG OUTPUT DEVICE
MICROPHONE
SPEAKER

DESIGN 1

Figure 2

2. the standard computer terminal performs fewer operations outside the computer demanding more computer time in the processing of student information.


RANDOM-ACCESS PLAYBACK UNIT
SOUND GENERATOR DIGITAL TO ANALOG
SPEAKER
MICROPHONE
COMPUTER
TERMINAL
DISPLAY SCREEN
KEYSET
ANALOG TO DIGITAL PROCESSOR
KEYBOARD

DESIGN 2

Figure 3

By way of explanation of the two schematics (Figures 2 and 3) the complex terminal has only two lines to the central computer since all audio information is processed by the special "music" terminal. The keyboard in the model is designed to judge the input of keyed "notes" and also activate the D to A output so that the user can hear the "performance" in real time. The student terminal also controls a playback unit of tape recorded information or music. The terminal, through the use of a processor, is capable of converting sound into digital information by means of an analog to digital unit. The final component in Design 1 is the digital to audio output unit, again under the control of the computer terminal.

The advantage of a complex terminal is the limited amount of computer processing time required to maintain user interaction over an extended period of time. The lack of data lines and communication needed between the terminal and the central computer is also a major asset.

The disadvantage of Design 1 is the cost of developing a unit with such capabilities. Each student who would work at the computer would need a terminal, with no sharing of the expensive A to D and D to A equipment.

By moving much of the "sound" equipment to the central computer area, the need for separate equipment is diminished in Design 2. In reviewing the diagram of a music system based upon a standard terminal and limited peripheral equipment, one can see the added difficulty of communication and time-sharing. The input and output lines to the computer are tripled in this design. With only one connection from the keyboard to the terminal, all other information must be sent to the computer for general processing, then must be returned to the individual user either visually on the terminal or aurally through a speaker/headphone system. Multiplexing input and output data on such a system for more than a few users becomes a major problem. By centralizing the sound treatment of the A to D and D to A information, data transfer is increased and consequently time-sharing becomes more of a problem on the larger scale computer system with many users.

The two designs can both be built to function effectively for the CBM system needs. Variations in and between these two designs involve the size of the system, size of the computer, and the sophistication of the analog to digital and digital to analog processors. In microcomputer terms, viewing either design for a small computer and one user, the designs become practically identical. By attaching another large computer to Design 2 for large core memory, Design 2 approximates Design 1, with Design 2's computer becoming a micro-processor.

In analysis of usage and needs, Design 1 is a logical choice for the large computer system with large numbers of users. Design 2 can effectively be used as a stand-alone computer system for use in a given location by a limited number of users.

The question of cost which is constantly with the designer and the user of a computer system is not an adequate determinate for the development of either design or the preference of either, for that matter. The fact remains that CBM systems are expensive today. Either design can be developed for existing computer systems for approximately the same cost. With interest in music as an area for computer research and development, the necessary hardware costs must decline with usage and availability of "standardized" integrated circuits. These designs can be constructed less expensively today than five years ago. This trend in hardware cost reduction should continue in the future as computer use increases steadily.

THE INVENTORY:

Computer technology dates from 1940 with the development of the high-speed electronic computer. Computer-assisted instruction dates from 1960, with the development and involvement of CAI in music dating from 1967. As we move toward a "complete" CBM system, care must be taken to develop the necessary hardware and software to aid the musician and the music student, not hinder their progress with an added technical learning burden.

The development of successful systems to date indicates that a quality hardware system can evolve which is simple and manageable for the user, while technically very complex in function. The development of hardware alone will not guarantee the successful application of computers to music performance of music instruction.

In the consideration of a total "complete" system, the software developed to utilize the hardware will make or break the CBM concept. One can easily list equipment that musicians and music students could enjoy in the classroom and studio. One can see the day-to-day development in allied technical fields which is solving or simplifying the hardware and software development problems in music analysis and instruction. What will remain is the task of developing strong effective software programs to utilize the "complete computer-based music system."

G. David Peters
School of Music
University of Illinois/Urbana/Champaign
Urbana, Illinois

GUIDELINES FOR THE DEVELOPMENT OF COMPUTER-ASSISTED INSTRUCTION MATERIALS IN MUSIC THEORY

Michael Arenson
Coordinator of Lower-Division Music Theory
Iowa State University, Ames

In the Spring and Summer of 1973, three groups of students enrolled in the School of Music at The Ohio State University proceeded through a computer-assisted instruction (CAI) program on musical structure based on the following ideas:

(1) The interrelationships of the various elements of music, such as sound, harmony, melody, and rhythm, are important in producing form.
(2) Music is an ongoing entity, existing as sound rather than as notation. Thus, all analytical observations of form take into consideration matters relating to performance.
(3) Valuable information about a piece of music can be gained by looking at the details as well as at the broader aspects of a piece of music. An analyst should examine the various dimensions of a piece of music.

The musical structure program was produced as part of a dissertation project concerned with the testing of a model for the systematic revision of CAI materials in music theory. As a result of an extensive examination of the kinds of errors that students made as they went through the program after each group tryout of the program, the author of the program made revisions. In addition he produced a list of eighteen guidelines that may be useful to others in producing effective CAI materials. Each guideline is the result of a specific concern that the author of the program had as he was systematically revising the program. The following is a list of the guidelines:

(1) The author must be open to criticism and suggestions of students and technical experts. If content order or procedure are not correct, rather than becoming defensive, the author should consider the criticisms and, if they make sense, provide whatever corrections are necessary.
(2) It is often the case that authors cannot see the logic behind students' unanticipated but reasonable responses. The student's thought process is often different from that of the author. However, careful tabulation of student comments can often reveal the ways students are thinking so that when a student response is reasonable but is judged incorrect by the author, the author can reconstruct the item to insure that that kind of response is no longer elicited.
(3) Students are often not very good typists. So the author should limit the required responses to a few characters or a few words at the most.
(4) There are a variety of ways that computer-assisted instruction systems provide editing functions, so that misspellings and parts of students' responses can be analyzed by the computer to see whether the student is answering basically correctly. Authors must utilize these functions and be careful to provide for the responses that students are likely to make.
(5) A good program is flexible. It often allows for a variety of correct-response options each of which has its own feedback. Often questions do not have one-and-only-one correct or meaningful answer or only one way of phrasing the answer when there is only one substantially-correct answer.

(6) Since learning-through-doing is important for programmed instruction, authors should provide the student with as many opportunities to respond as possible.

(7) One of the advantages of computer-assisted instruction over a programmed text is the ability of CAI to provide, without great inconvenience to the student, very helpful feedback for anticipated wrong answers. The author should provide good and plentiful feedback so that the student will learn from mistakes.

(8) The author should provide considerable review in a program. Students forget easily. This is even more imperative if the terminal is a cathode-ray-tube or PLATO plasma display where students cannot refer back to a printout.

(9) In programmed instruction in general, the use of prompts is important and the use of a known-to-unknown prompting situation works out well; it is good to introduce a term in one context and then to ask a question where that term is used in a related context.

(10) Terminology chosen for usage in a program must be exact--not loose. Because musicians have borrowed terms from other disciplines such as physics, authors of music programs must be especially careful to to use only terms which really describe what they are talking about.

(11) In programmed instruction, as in most educational media, thorough explanations are imperative. When concepts are introduced for the first time, the author must make clear what the relationships are between those concepts and earlier related ones.

(12) When having a student refer back to previous material studied, the author must be specific about where the student is to look. A page number or other citation to a specific block of information is imperative. Otherwise, the student wastes too much time.

(13) In CAI generally and in music CAI specifically, the simultaneous use of a variety of media resources can help make the student more attentive to questions asked. The media may consist of some combination of items such as a cathode-ray-tube or Plasma screen, random-access slides, or tape recorder.

(14) It is important for authors who use accompanying materials to study carefully the flow between these materials and computer-stored materials regardless of whether the accompanying materials are reading materials or items such as slides, audio materials or motion pictures.

(15) All accompanying materials must be organized so that students can easily and quickly refer to those materials.

(16) When using a variety of different kinds of materials in a program, an author must be careful to make numbering systems and points of reference clear in those materials.

(17) Too much textual material is bad in CAI; CAI cannot be simply a page turner. When much textual material is required, the author should provide packets of textual material at the terminal site. If textual material is necessary in a program, the author should provide a large number of places where the terminal stops printing or displaying text so that the student can read small bits of material before receiving more material.

(18) There is great need for graphic terminals in areas such as music, so that students can have a large range of convenient interactions while engaged in a CAI program. For example, with graphic terminals, students can work directly with a musical staff. Such interaction with a staff is not possible with the symbolic limitations of typewriter characters.

The eighteen guidelines above are by no means an exclusive list but do represent some of the most important items about which an author of a music CAI program should be concerned in generating an effective program.

LIST OF REFERENCES

Abedor, A. J. Development and Validation of A Model for Formative Evaluation of Self-Instructional Multi-Media Learning Systems. Paper before the 1972 Annual Meeting of The American Education Research Association.

Cone, E. T. *Musical Form and Musical Performance*. New York: W. W. Norton and Company, Inc., 1968.

Hartley, J. *Strategies for Programmed Instruction: An Educational Technology*. London: Butterworths, 1972.

LaRue, J. *Guidelines for Style Analysis*. New York: W. W. Norton and Company, Inc., 1970.

Markle, S. M. *Good Frames and Bad; Grammar of Frame Writing*. New York: John Wiley and Sons, Inc., 1964.

Reed, F. C., Ertel, P. Y., & Collart, M.E. A Model for the Development of Computer Assisted Instruction Programs. *Educational Technology*, 1974, *14*(3), 12-20.

Interactive Simulation/Games as an Aid to Musical Learning*

Fred T. Hofstetter
University of Delaware

- Abstract -

Whereas interactive simulation/games have been successfully used for years in computer-based instructional systems, they have received little attention in music. There is a great potential for in-depth learning of music to take place through interactive simulation/games. An analysis of instructional needs suggests that the proper application of simulation/games is in the basic skills area of the music curriculum. Examination of available hardware indicates that this application can be supported by today's technology. Designs are presented which deal with chords, harmonies, rhythms, timbres, intonation, error detection, and error correction. Finally, it is suggested that interactive simulation/games may be an important new direction for computer-based musical instruction.

- Introduction -

Examination of the current state of computer-assisted instruction in music (Jones, 1975) shows that developers have concentrated on tutorials and drill-and-practice exercises as their main instructional methods. Computer-based tutorials and drills have proven to be quite effective in improving student learning as documented in studies by Diehl (1973), Hofstetter (1975), Peters (1975), and Placek (1974). The pedogogical research potential of reports on student learning patterns obtained by these methods has been demonstrated by Killam, Lorton, and Schubert (1975). The success of these efforts motivates developers of computer-based musical materials to expand the repertoire and extend the scope of tutorials and drills.

There are two other instructional methods which have received relatively little attention in music, but which are fairly widespread among other disciplines. These methods are the computer-based game and simulation. Games and simulations are related in that both methods use the computer to place the student in a situation which requires thoughtful manipulation of some parameters in order to achieve a desired result. Twelker and Layden (1973) have given a practical division of the two methods by stating that games feature competition among individuals or groups, whereas simulations emphasize placement of the individual in a

* ©1977 by Norlin Music Inc.

real-life situation. Methods which both place the individual in a real-life situation and involve competition are called simulation/games.

Peter Katz (1975) has drawn a cartoon which shows a student and a teacher arguing about a CAI assignment. The student maintains that since the assignment is fun, it is a game. The teacher argues that because the assignment is educational, it is a simulation. Simulation/Games are both educational and fun. It is the purpose of this paper to show how simulation/games can be integrated into the basic skills area of the music curriculum. The use of simulation/games will not only provide music students the opportunity of developing their basic skills in a simulated performance environment, but it will also increase their motivation by transforming the practice of basic skills from a chore into a fun activity.

- Analysis of Instructional Needs -

The consideration of the use of simulation/games must begin with an analysis of the instructional needs in the basic skills area. Musicians take on a variety of roles, including those of composer, conductor, performer, historian, theorist, therapist, and teacher. Each of these roles is complex in that the success of the musician largely depends on mastery of skills needed to be effective. Those requisite skills which are common to all musicians are called basic skills. The operational definition of these skills can be given as follows: in order to be successful, the musician must develop an eye that hears and an ear that sees. In other words, musicians must be able to look at a musical score and know how it will sound, and conversely, upon hearing music they must know how it should be notated. It is this basic skill that makes it possible for a conductor to detect and correct errors in musical performance, for a composer to tell performers what notes they should play, for a teacher to tell how well students are learning, and for the critic to make judgments regarding musical style.

The traditional approach to training musicians in basic skills has relied almost exclusively on tutorials and drills. Musicians are taught the elements of musical notation and syntax in a tutorial format, and the aural recognition of those elements is taught by means of dictation drills whereby music is played by the instructor, and students attempt to write down what is played. The problem with this approach is that the student is never placed in the situation of a real musician. The assumption is that if students master the drill materials, they will be able to correctly make the aural judgments required of a musician. However, evaluations by both teachers and students have shown that this approach is not adequate. There is a need for the tutorials and drills to be supplemented by role-playing activities in which the students can practice the basic skills under the supervision of a teacher in real-life musical situations.

Attempts to provide basic skills training in a real-life environment have been hampered by a prohibitive managerial overhead. First, allowing students to learn in real-life situations requires that teachers work with them on an individual basis, and the workload of the teacher is already too full in the traditional classroom approach. Second, the cost of providing performances for the students to work with would be prohibitive. Third, the instructional space needed to accomodate the real-life approach would be many times what it is in more traditional instruction. What is needed is a device which can simulate real-life musical situations at a reasonable cost without creating an unmanageable administrative overhead.

- Examination of Available Hardware -

The requirements of simulation/game applications are as follows. There must be a screen capable of displaying musical scores. The display must be touch-sensitive so that the student can point out errors in performance by touching the screen. The display must be selectively erasable so that the student can correct errors which might be written in the score. The system must be able to randomly access pre-recorded musical compositions. The audio component must also be capable of synthesizing music in real time, and of simulating instrumental sounds. The overall response time of the system should be so fast that the simulated performing group would immediately respond to the wishes of the student.

At the present time no system exists which can provide all of these hardware requirements. However, it is technologically feasible to engineer such a system. All of the necessary components do exist, although some are more highly developed than others. The plasma terminals used on the Control Data Corporation's (1976) PLATO system are capable of displaying musical notation either by drawing it or by projecting it by means of a rear-screen microfiche projector. The screen is both touch-sensitive and selectively erasable. The PLATO system also provides very fast response times whereby the student could work with simulated performing groups in real time.

A suitable random access audio device has yet to be developed. However, there are two on the market which have potential for the future, and which can be used in prototype systems. The first is the RA-4096 random-access audio recorder manufactured by Education and Information Systems, Inc., (1976). This device employs a movable record/playback head which randomly accesses pre-recorded audio disks. Each disk can hold up to 22 minutes of music. The access time is excellent, an average of .5 second. The disadvantage of this machine is that the frequency response is only 120 hz. to 6000 hz. There are not enough high frequencies to give a faithful reproduction of instrumental sounds. The second device is the Phi-Deck computer-controllable cassette tape transport manufactured by the Triple I (1976) division of the Economy Company. This machine gives faithful sound reproduction throughout the audible frequency range. Its drawback is that its access time is not as quick as that of the audio disk. The average access time of the Phi-Deck is estimated to be 10 seconds for musical simulation applications.

Techniques of digital synthesis have advanced to the stage at which digital synthesizers are beginning to make realistic simulations of musical performance. The Dartmouth music system created by Alonso, Appleton, and Jones (1975) contains a real-time digital synthesizer which provides sophisticated control over all aspects of sound. This system will soon be available from the Norlin Music Company which will market it for educational and compositional purposes. Other developments in this area are reported in the abstracts of the First International Conference on Computer Music organized by Vercoe (1976). A simpler, low-cost digital synthesizer has been developed by Sherwin Gooch at the University of Illinois' Computer-Based Education Research Laboratory. Called a "Music Box," this four-voice device provides control over frequency, time, and loudness, but not timbre. The four voices are sawtooth waves. The music box has been successfully used in a variety of applications, and as indicated in the next part of this article, it can also support some simulation/games.

- Applications of Simulation/Games -

Simulation/Games can be designed for one player, two players, or groups of several players. Their degree and kind of competition can vary. It is the purpose of this section of the article to describe a proposed package of six simulation/games which place students in a variety of situations with instructional emphasis on intervals, melodies, chords, harmonies, rhythms, timbres, intonation, error detection and error correction. Figure 1 shows the titles and classifications of the games in the order in which they will be discussed.

1. Go to the Head of Your Class. This game is highly competitive, and it involves all students in a given class. It is intended to be an integral part of drill-and-practice dictation programs dealing with intervals, melodies, chords, harmonies, and rhythms. The instructor can allow students to play this game either as a reward for having completed a specific dictation learning task, or as a student-initiated branch in a dictation program. This game is played by one student at a time. The object of the game is for the player to correctly answer a set of at least ten computer-generated dictation questions faster than anyone else in the class. The questions deal with one kind of dictation, that is, either ten intervals, melodies, chords, harmonies, or rhythms, depending on the kind of dictation which the student does prior to beginning the game. The computer maintains class rosters for each kind of dictation. When a student completes the game, the average time it took to answer a question is recorded next to the student's name in the class roster. The student is informed of his rank in the class, and is given the option of seeing the entire class roster in rank order. In this game the lowest score wins.

The algorithm used to compute the player's average response time allows the student a lot of freedom in choosing the format in which questions are asked, while at the same time it preserves the overall integrity of the ranking system. The student is allowed to select the length of time the stimulus will be played and the number of times the stimulus will be

Title	Instructional Emphasis									Format	
	Intervals	Melodies	Chords	Harmonies	Rhythms	Timbres	Intonation	Error Detection	Error Correction	Competitive?	Number of Students
Go to the Head of Your Class	X	X	X	X	X					Yes	All students
Ear War	X									Yes	2
Musical Hangman		X								No	1
Timber!						X				No	1
Cry Wolf							X		X	No	1
Lully's Baton								X	X	No	1

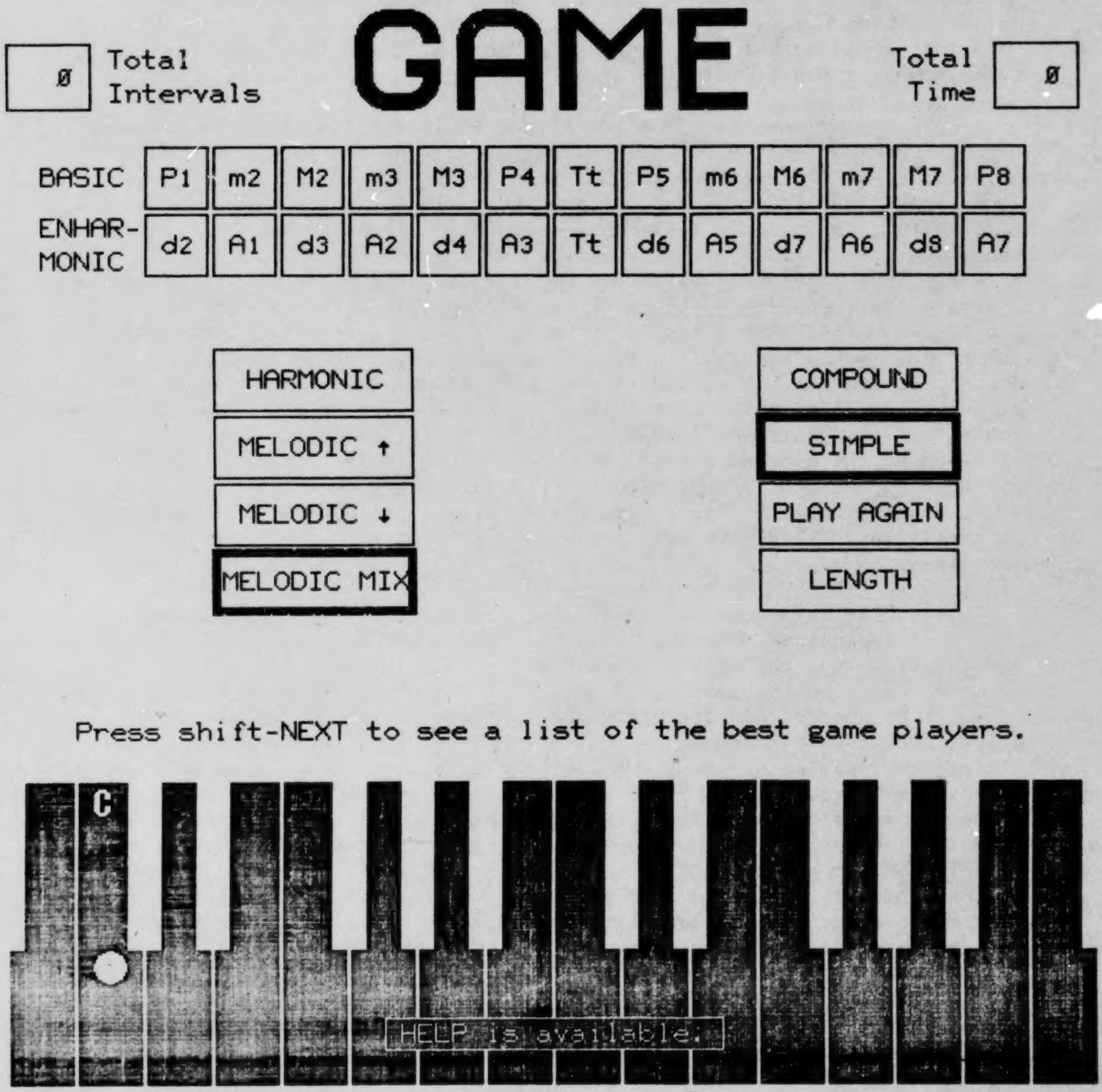
Figure 1. Classification of Simulation/Games

repeated. There is no limit placed upon the number of responses the student can try before arriving at the correct answer. The student's score is computed as $(Q + L) \div N$, which is the amount of time spent answering questions (Q) plus the amount of time spent listening to stimuli (L) divided by the number of questions (N). This algorithm encourages students to select a stimulus length as short as possible, and to avoid requesting repetitions of the stimulus in order to achieve the best score.

At the University of Delaware, the implementation of this game has begun as part of the GUIDO ear-training package on the PLATO system. The intervals game is completed and has been on the system for about a year. It has generated a lot of enthusiasm, and trying to get a high rank on the game roster has become a real contest in the music department. Figure 2 shows the game board as it is displayed on the touch-sensitive PLATO screen. When intervals are played by PLATO's Gooch music box the students indicate their answers by touching the appropriate square at the top of the screen. The boxes drawn in the middle of the screen are used to alter the manner in which questions are asked. The keyboard is available as a diagnostic aid which the student can touch in order to match the pitches of intervals being played. Figure 3 shows the first page of the class roster as it appeared on January 1, 1977. The first column shows the ranks attained by the game players. Column two gives their sign-on names followed by the name of their course. Column three shows the average time in seconds it took them to answer the intervals followed by the number of intervals they tried. Column four shows how many days they held rank in the class.

2. Ear War. Ear war is a simulation/game for two players, each seated at a separate terminal. Similar to PLATO's dogfight game (Green and White, 1976), the object of ear war is to shoot down one's opponent by maneuvering on the game board such that the opponent comes within range of one's armaments. In the PLATO dogfight game, each player has an airplane. The computer terminals display the field of play, showing to each player the location of the planes. The players move the planes by pressing directional arrows on the PLATO keyset. These arrows move the planes up, down, across, and diagonally on a dot matrix. Dogfight is a purely recreational game. It is very popular, having been played over half a million times. Ear war adapts this very popular game for instructional purposes by changing its game board into a musical matrix, and by requiring the players to listen to and identify their desired moves which are heard as musical intervals.

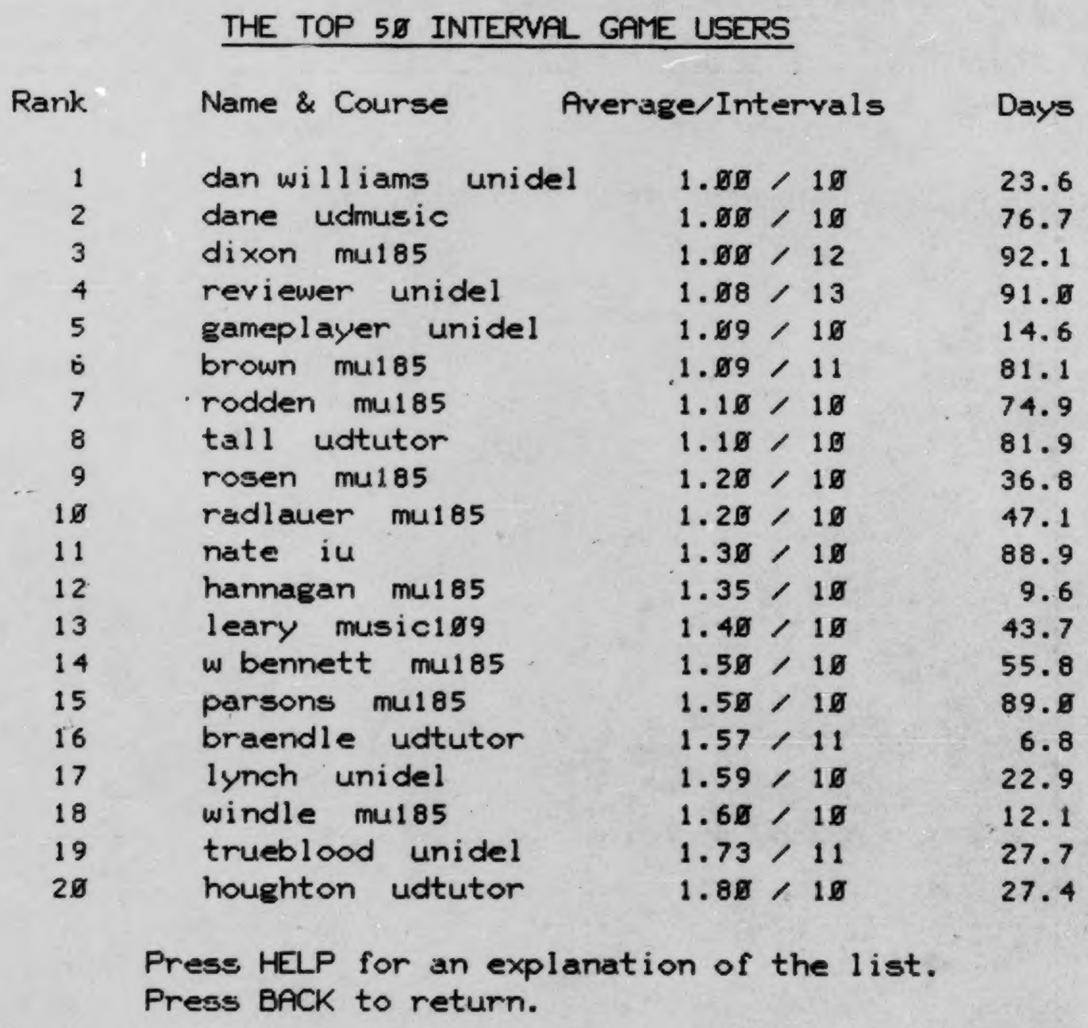
Figure 4 shows the game board for ear war. It contains a dot matrix which shows the positions for each player. When ear war begins, the computer randomly generates a set of pitches which are overlaid across the top and down the left-hand side of the matrix. The players do not get to see these pitches. However, when they attempt to move on the matrix, the two notes represented by the dot to which they wish to move are played for them, and before the move is allowed they must correctly identify the interval which the two notes form by touching the appropriate interval square at the top of the matrix. When a player comes within range and


GAME
0 Total Intervals
Total Time 0
BASIC
P1 m2 M2 m3 M3 P4 Tt P5 m6 M6 m7 M7 P8
ENHAR-MONIC
d2 A1 d3 A2 d4 A3 Tt d6 A5 d7 A6 d8 A7
HARMONIC
MELODIC ↑
MELODIC ↓
MELODIC MIX
COMPOUND
SIMPLE
PLAY AGAIN
LENGTH
C

Press shift-NEXT to see a list of the best game players.

HELP is available.

Figure 2. Touch Sensitive GUIDO Game Board. © 1976 by the University of Delaware. Programmed by William H. Lynch.



```
                 THE TOP 50 INTERVAL GAME USERS

Rank       Name & Course        Average/Intervals       Days

  1        dan williams  unidel       1.00 / 10         23.6
  2        dane  udmusic              1.00 / 10         76.7
  3        dixon  mu185               1.00 / 12         92.1
  4        reviewer  unidel           1.08 / 13         91.0
  5        gameplayer  unidel         1.09 / 10         14.6
  6        brown  mu185               1.09 / 11         81.1
  7        rodden  mu185              1.10 / 10         74.9
  8        tall  udtutor              1.10 / 10         81.9
  9        rosen  mu185               1.20 / 10         36.8
 10        radlauer  mu185            1.20 / 10         47.1
 11        nate  iu                   1.30 / 10         88.9
 12        hannagan  mu185            1.35 / 10          9.6
 13        leary  music109            1.40 / 10         43.7
 14        w bennett  mu185           1.50 / 10         55.8
 15        parsons  mu185             1.50 / 10         89.0
 16        braendle  udtutor          1.57 / 11          6.8
 17        lynch  unidel              1.59 / 10         22.9
 18        windle  mu185              1.60 / 10         12.1
 19        trueblood  unidel          1.73 / 11         27.7
 20        houghton  udtutor          1.80 / 10         27.4

        Press HELP for an explanation of the list.
        Press BACK to return.
```

Figure 3. First Page of Interval Game Roster. © 1976 by The University of Delaware. Programmed by William H. Lynch.

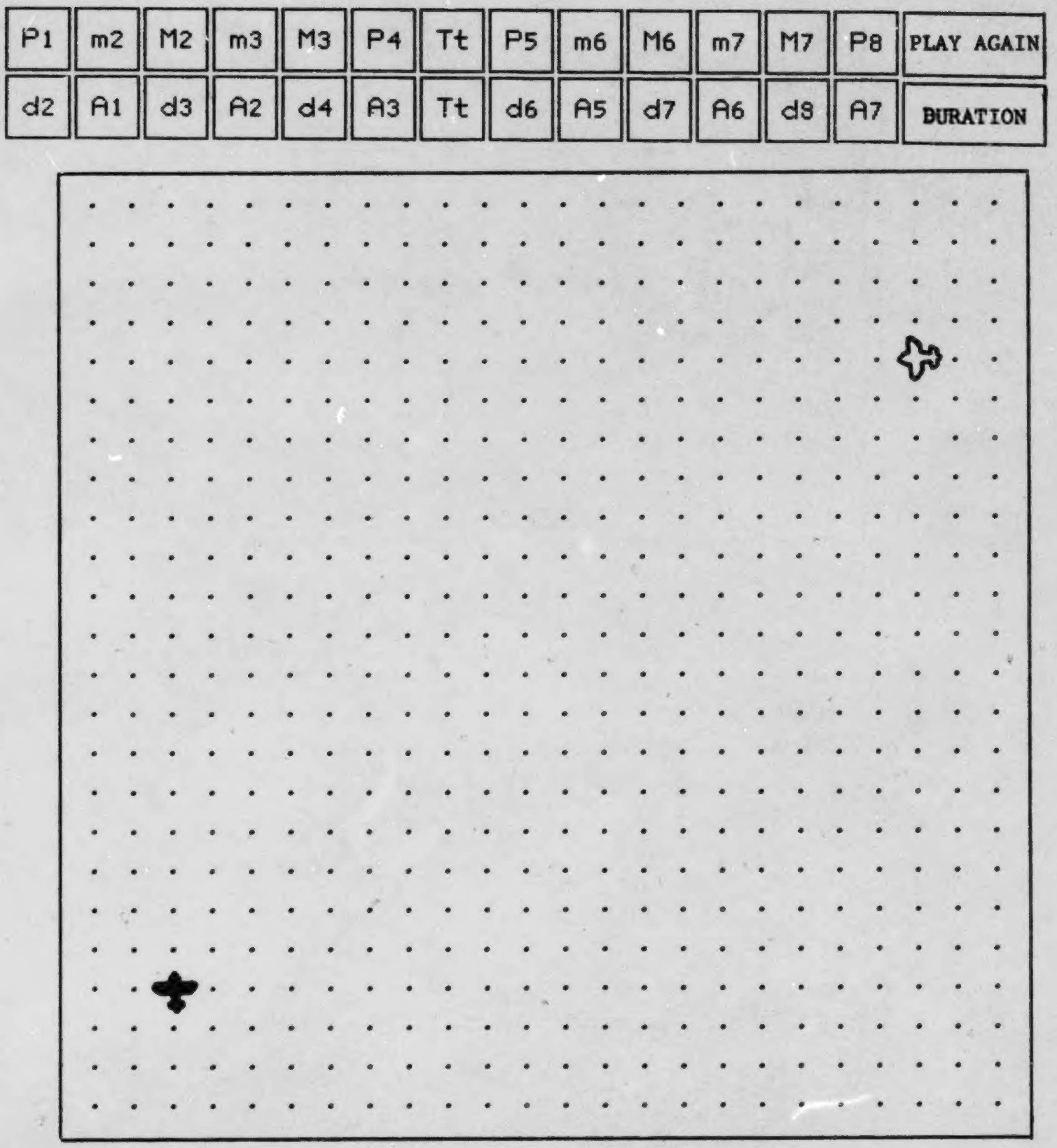

P1
m2
M2
m3
M3
P4
Tt
P5
m6
M6
m7
M7
P8
PLAY AGAIN
d2
A1
d3
A2
d4
A3
Tt
d6
A5
d7
A6
d8
A7
DURATION

Figure 4. Touch-Sensitive Display of Ear War.

attempts to shoot his opponent, he must first name the interval of the dot occupied by his opponent. If his opponent occupies that dot when he names it, he wins. It is anticipated that this game will be very popular among music students, and that it will make the learning of intervals a lot more fun.

3. *Musical Hangman*. Perhaps the most popular, the most available, and the most played computer-based instructional game is *Hangman*. In this game the computer "thinks" of a word and displays on the terminal screen as many blanks as there are letters in the word. The person playing the game guesses what the word is by typing in single letters which are thought to be in the word. When the player types in a letter which *is* in the word, the computer prints that letter in each blank in which that letter occurs. However, when the player types in a letter which is *not* in the word, the computer displays part of a computer graphic which depicts a hanged game player. Each time an incorrect letter is typed, the computer adds more to the picture of the hanged game player. The object of the game is to cortectly enter all of the letters in the word before the computer completes the picture, thereby hanging the player.

The design of the Hangman game is fairly simple, but its instructional benefits are far reaching. Through successive guesses of letters which might be in the word, the game player gains more and more information about the content of the word. When several correct guesses have been made, the basic structure of the word begins to be revealed. At this point the player chooses letters according to what he knows about the structure of the language. Playing Hangman arouses the player's curiosity about the language. The more Hangman is played, the more advanced the player's intuitive grasp of word structure becomes. Through experience the player becomes better at playing the game, and is hanged less often.

The musical adaptation of Hangman, called "Musical Hangman," works basically the same way as regular Hangman except that instead of having the computer think of a word, the computer thinks of a melody from music literature. The game player guesses notes which might be in the melody, and when correct guesses are made, the computer displays those notes on the terminal screen. In addition, a special feature is made available in musical hangman: the player can ask to hear the notes which have been displayed in their proper rhythmic placements, with unnamed notes played as percussive sounds. The player or the instructor can also set the level of difficulty as *easy* which contains short diatonic melodies, *average* which contains longer melodies with some chromatic notes, or *hard* which contains advanced melodies. At the end of the game, regardless of whether the computer hangman or the player has won, the composer and source of the melody are revealed, and the entire melody is played.

Aids to musical learning obtained from playing this game are as follows:

1) Students will become more aware of the motivic structure of

melodies because by guessing notes of the melody they will see how motivic patterns occur when the notes are displayed in their respective places.

2) Students will be helped in their study of musical literature by learning melodies from works by master composers.

3) Students will be more sensitive to keys, modes, and scales because one of the main ways to beat the hangman is to determine from a few correct guesses what keys, modes, and scales are used in the melodies.

4) Students will be more highly motivated to explore melodies in this manner because playing Musical Hangman is so much fun.

4. Timber. Musicians frequently refer to their performing instrument as their "axe." Lumberjacks use axes to fell trees. When they fell trees, lumberjacks yell "timber!" Musicians use a word spelled almost the same as "timber" to refer to the tone quality of their instruments, which is called "timbre." This game plays upon the double meaning of the word "axe" and the close spellings of "timber" and "timbre" in creating a scenario in which music students identify the sounds of individual musical instruments and their combinations.

"Timber" is a one-person, one-terminal game. The student is asked to pretend that he is a lumberjack attempting to chop down a tree. The computer informs him of the size and hardness of the tree, and tells him how many swings of the axe it will take to fell the tree. The player must earn each swing of his axe by aurally identifying the source of some instrumental sound. If the student gives a wrong answer, his swing misses the tree. If the student misses more than 25% of the swings needed to fell the tree, he loses the game and is told to sharpen his axe before trying another tree. Correct answers are rewarded through computer graphics which display the tree and record the effects of each swing of the axe until the tree is felled with a mighty "Timber!"

The implementation of this game requires the use of a computer-controlled audio unit capable of faithfully producing the sounds of musical instruments at a reasonable cost. At the present time the only device which can do that is the random-access cassette tape recorder, with its inherent slow access times. The technology of computer music is advancing to a stage in which it will soon be possible to generate instrumental sounds through digital synthesis. This will make it possible to faithfully simulate performing groups in real time, a capability essential to the implementation of the next two games.

5. Cry Wolf. When a musician plays a note on his performing instrument, he must not only press the proper levers, finger the right keys, and cover the appropriate holes, but he must also play that note in tune. If he is not holding his instrument at the proper angle, or if his embouchure is incorrect, or if he is not applying just the right amount

of pressure, the pitch of the note can be a little too high or too low. Errors in tuning are called "wolf tones," and it is crucial for a musician to be able to tell when a note is out of tune.

When playing "Cry Wolf," the student would imagine himself to be a conductor. The computer would display a musical score on the terminal screen, and it would then play the score, randomly introducing out of tune notes in the performance. The student would be asked to identify and correct the wolf tones by touching them in the score and indicating whether they are flat or sharp. However, instead of telling the student whether each correction has been properly made, the computer lowers the notes which the student says were sharp, raises the notes said to be flat, and plays the result. Thus the student can make things worse instead of better, as would be the case in a real musical performance. In "Cry Wolf," the student must keep trying until all of the notes are in tune.

6. Lully's Baton. Jean-Baptiste Lully, an Italian born French composer, was one of the most colorful conductors of the seventeenth century. He provided us the model from which our modern orchestra developed, and he made famous the use of the baton for giving an orchestra the beat. The circumstances of his death in 1687 provide the background for this simulation/game, because they show how important it is for a conductor to be able to identify and correct errors made by individual players. Schonberg (1967) reports how Lully worked himself into such a rage that while rehearsing his orchestra he rammed his baton through his foot. Gangrene set in, and Lully died.

It is hoped that by playing Lully's Baton students can gain enough experience in a simulated musical environment that they will not share the fate of Jean-Baptiste. In this game the computer displays musical scores and simulates errors in instrumental performance as in "Cry Wolf." The difference is that whereas in "Cry Wolf" the students were asked to correct intonation errors, in "Lully's Baton" they are asked to correct note-reading errors related to pitch and rhythm. As in "Cry Wolf" the students can change any notes in the score, and thus they could make things worse instead of better. Upon successful correction of all errors in a given performance, the students are confronted by the ghost of Lully who says: "Bravo!"

- Conclusion -

This paper has introduced the simulation/game as a potential aid to the teaching of basic musical skills. Of the six models suggested for its application, only the first has been implemented. However, the success of this first attempt coupled with the widespread benefits achieved in other disciplines indicates that the simulation/game will be a fruitful new area for instructional developments in music. The new Norlin Music Education System will contain all six of the games described in this paper.

References

Alonso, S., J. H. Appleton, and C. A. Jones. A special purpose digital system for the instruction, composition and performance of music. *Proceedings of the Sixth Conference on Computers in the Undergraduate Curriculum*, 1975.

Control Data Corporation. *CDC PLATO terminal user's guide*. St. Paul: Control Data Corporation, 1976, CDC Publication Number 9740800.

Diehl, N. C. and R. H. Ziegler. Evaluation of a CAI program in articulation, phrasing, and rhythm for intermediate instrumentalists. *Council for Research in Music Education*, 1973, 31, 1-11.

Education and Information Systems, Inc. Introducing the RA-4096. 1976 Brochure available from EIS, 802 West University, Urbana, Illinois, 61801.

Green, Douglas Jay and Lawrence A. White. *Dogfight*. Urbana: University of Illinois PLATO System, 1976. PLATO IV computer program.

Hofstetter, F. T. GUIDO: An interactive computer-based system for improvement of instruction and research in ear-training. *Journal of Computer-Based Instruction*, 1975, 1, 100-106.

Jones, M. *Computer-Assisted Instruction in Music: A Survey With Attendant Recommendations*. Ph.D. dissertation. University of Iowa, 1975, 113-114.

Katz, Peter. Simulation Cartoon. In: *What to do after you hit return or P.C.C.'s first book of computer games*. Menlo Park: People's Computer Company, 1975, p. iii.

Killam, R., P. Lorton, and E. Schubert. Interval recognition: A study of student accuracy of identification of harmonic and melodic intervals. *Journal of Music Theory*, 1975, 19(2), 212-234.

Peters, G. D. *The Development of Computer-Assisted Instructional Materials for Use in the Teaching of Instrumental Music via PLATO IV*. Final Report, Undergraduate Instructional Award, University of Illinois, 1975.

Placek, R. W. Design and trial of a computer-assisted lesson in rhythm. *Journal of Research in Music Education*, 1974, 22, 13-23.

Schonberg, Harold C. *The great conductors*. New York: Simon and Schuster, 1967, pp. 35-36.

Triple I Division of the Economy Company. Phi-Deck. 1976 Brochure available from Triple I, Box 25308, 1901 North Walnut, Oklahoma City, Oklahoma 73125.

Twelker, Paul A. and Kent Layden. A basic reference shelf on simulation and gaming. In: *The Guide to Simulations/Games for Educational Training*. Lexington: Information Resources, Inc., 1973, pp. 445-455.

Vercoe, Barry, Ed. *Abstracts of the first international conference on computer music, October 28-31, 1976*. Cambridge: Department of Humanities, Massachusetts Institute of Technology, 1976.

A COMPUTER-MANAGED INFORMATION AND INSTRUCTIONAL SYSTEM FOR VOCATIONAL AND TECHNICAL EDUCATION

by

Nevin R. Frantz, Jr. and Edward E. Boas, Jr.
Occupational Education Department
University of Delaware

The advent of computer-based technology has produced various applications in the administrative and instructional functions of secondary and post-secondary vocational and technical education. Several projects have dealt with the use of computers in managing instruction. Rannells (1973) described the use of a computer in evaluating competencies students acquired in an individualized instructional program for horticultural occupations. A modular approach which utilized a computer to record student achievement of job tasks was discussed by Dupuis and Bell (1974). A computer-managed system was developed by Danford (1974), which prescribed instructional modules, scheduled students, and recorded their completion of the modules. These computer-based projects were limited to single campus applications and emphasized collecting and recording student-learning task assignments. At the present time, in the field of vocational and technical education, there is a lack of activity in investigating the feasibility of disseminating curriculum information and managing the instructional process on a multi-campus basis. An investigation is needed that would examine the possibility of disseminating curriculum information and managing instruction in order to alleviate several problems facing instructors and students in vocational and technical education.

Curriculum Development and Instructional Management Problems. Instructors employed in vocational and technical education institutions at the secondary and post-secondary levels are constantly in need of information about available curriculum materials. These teachers are continuously involved in the process of constructing course syllabi, developing instructional objectives, identifying learning resources, and preparing evaluation instruments. In many instances, much of the needed material for a particular program is already in existence, but its availability is usually unknown to the instructors. Although the establishment of the ERIC system and several regional curriculum centers for vocational and technical education have been positive efforts in overcoming the problem, the fact remains that most instructors are not aware of these agencies nor do they have the time available to retrieve existing material from these sources. A system is needed that could retrieve and disseminate curriculum information expediently to instructors in local educational agencies.

Recent interest in the development and utilization of individualized instructional modules with students in vocational and technical education programs has also created logistical and instructional management problems. Instructional modules prepared by state and local school curriculum development projects as well as individual teacher efforts are often not available to other interested instructors. The logistics and costs of reproducing, storing, and disseminating large quantities of prepared instructional materials become insurmountable and lead to widespread duplication of curriculum development effort within a state or region, as well as throughout the nation. The mechanics of field testing, revising, and updating prepared instructional modules is time consuming and requires costly reproduction which is also inefficient and impractical.

Instructors utilizing instructional modules with students have encountered problems in collecting and recording individual student entry behaviors, instructional preferences, and achievement progress. Traditional methods of managing instruction create a situation where instructors become "record keepers" which is not personally fulfilling and frustrating to students due to a reduction in teacher-student interactions. Perhaps a computer-managed system could be utilized in delivering instructional modules to students and collecting, processing and supplying information on each student for the instructor. It was the purpose of this project to develop: (1) a model which would provide a curriculum materials dissemination system for teachers located on various campuses of a vocational and technical education system and (2) a computer-based system to deliver and manage instruction on an individual basis for students enrolled in several program areas on various campuses.

Curriculum Information System. The development and field testing of the computer-managed information system was accomplished through funding from Part C of the 1968 amendments to the Vocational Education Act of 1963 and the services provided by membership in Project DELTA. Project DELTA, funded since 1972 by the Delaware School Auxiliary Association, is devoted to "research activity which is dedicated to the establishment of a computer network designed to provide a computer use educational facility." The project utilizes a DEC PDP 11/50 system housed at the University of Delaware with 35 remote Decwriter II terminals located in various schools throughout the state (See Appendix A.)

In order to provide a curriculum information dissemination system, a matrix was prepared which categorized curriculum materials according to their type and relationship to an occupational program area. The categories ranged from field trips and guest speakers to instructional objectives and criterion referenced measurement instruments. The occupational program areas were based upon the fifteen U.S. Office of Education Career Clusters which were futher divided into sub-clusters and specific occupation areas (See Appendix B.)

The required programming was completed and a data base was assembled using the PDP 11/50 system. The curriculum information system was field tested with a group of instructors in one school who could access the information according to needs for curriculum information in their occupational areas. Preliminary results indicated the approach was feasible and current efforts are being made to utilize the system in disseminating, on a state-wide basis, community resources such as guest speakers and field trips, to instructors who are incorporating career education concepts into their subject areas.

Instructional Management System. This was established by developing the instructional module category of the curriculum information dissemination system. In addition to the information provided, a series of instructional modules were prepared and assembled in this category. These modules consisted of a precheck, a terminal performance objective, several enabling objectives with listed instructional resources, prescribed student activities, feedback, and a post-check.

The completed computer-managed instructional system was field tested with a group of ninth grade students enrolled in several program areas related to the transportation cluster at Delcastle Vocational-Technical High School. A small number of students in these programs were given the opportunity to interact with a computer terminal in selecting instructional modules appropriate for their career interest and tentative occupational preparation plans. When each student selected their respective modules, this information was processed and a printout obtained which scheduled them for a teacher, classroom, and time period. The next procedure was to begin the first instructional module. If a student felt that they had already acquired some knowledge or skill needed to perform the terminal performance objective of the instructional module they received a precheck from the computer which appraised their entry behavior. Their responses to the precheck were entered into the computer which analyzed the information and provided a printout which indicated which instructional package(s) the student would use. If the student did not take the precheck, the instructional packages were then provided on the terminal.

After receiving the instructional package printouts (See Appendix C), the students moved to their respective laboratories. In using the instructional package, each student initially read the enabling objective and selected learning resources from those listed which are considered appropriate for their learning style. After utilizing the resource, the student applied the knowledge or practiced the skill as directed in the activity section. Upon completion of the learning activity, each student verified their performance in the feedback or check section. If the criterion performance as stated in the enabling objective was accomplished successfully, the student advanced to the next instructional package. Students who were unsuccessful turned to a recheck or remedial section and completed the activities as directed until mastering the enabling objective.

After successfully completing the enabling objectives for all instructional packages, the student was evaluated on achievement of the terminal performance objective. The student returned to the computer terminal and obtained a post-check which contained multiple choice items and/or directed the student to a skill performance activity which the instructor evaluated. After completing a multiple choice post-check, the student recorded his responses at the terminal and received feedback concerning his status. If the result was satisfactory, the next instructional module would be obtained by the student. Students who were not successful would be directed by a printout to engage in a remedial route such as completing a specific activity before attempting the post-check again. The status of each student's location in the instructional sequence was automatically recorded and delivered upon request to teacher, counselors, and administrators. (See Appendix D.)

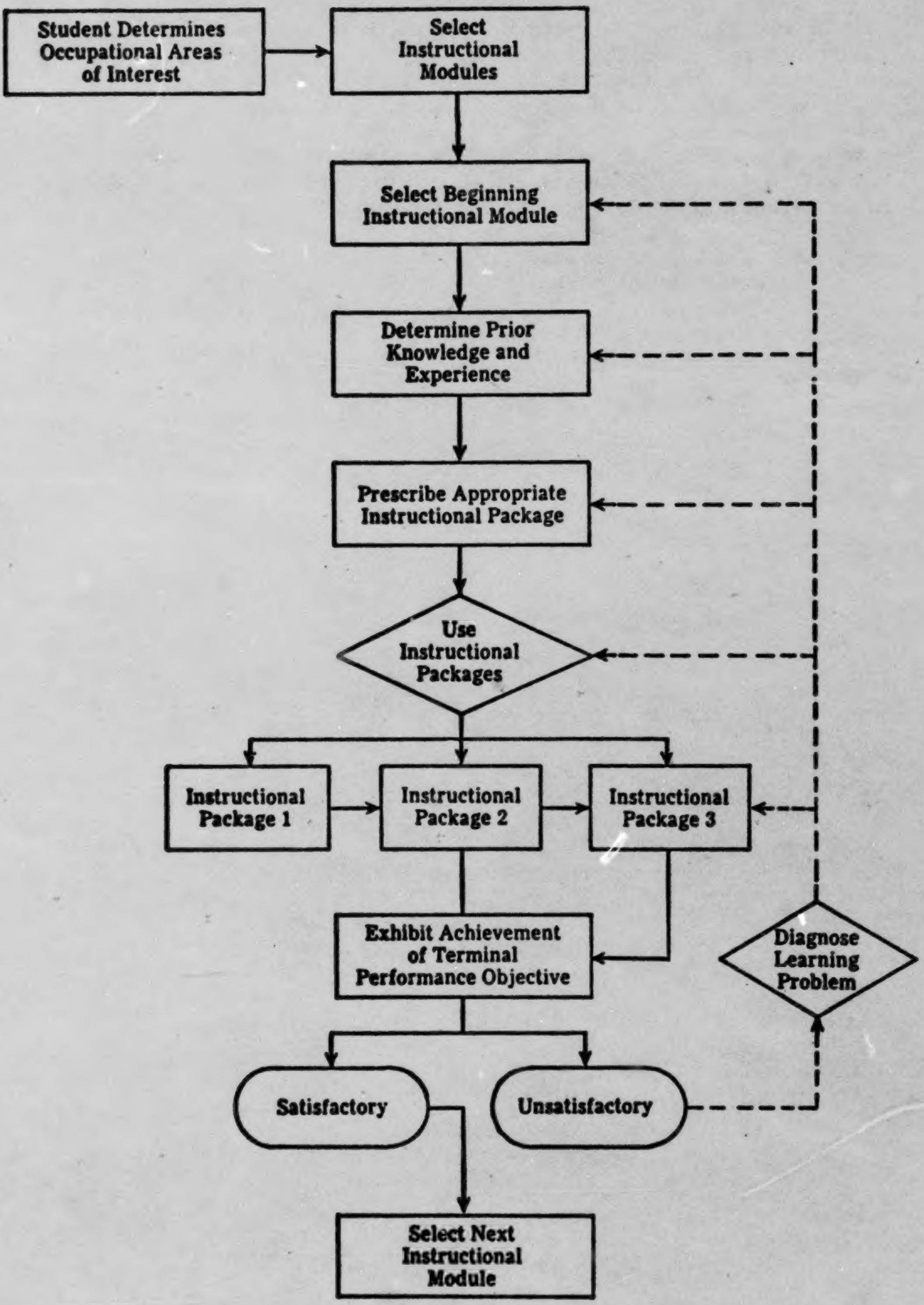

Student Determines Occupational Areas of Interest
Select Instructional Modules
Select Beginning Instructional Module
Determine Prior Knowledge and Experience
Prescribe Appropriate Instructional Package
Use Instructional Packages
Instructional Package 1
Instructional Package 2
Instructional Package 3
Exhibit Achievement of Terminal Performance Objective
Diagnose Learning Problem
Satisfactory
Unsatisfactory
Select Next Instructional Module

Figure 1. Computer Managed Instructional System

In conclusion, the computer-managed instructional system for vocational and technical education discussed in this paper was field tested in a pilot project using a small number of instructors and students in one occupational cluster (transportation). Although initial results in developing and implementing the system have been successful, additional activities are needed to expand the approach and evaluate the effectiveness and efficiency of utilizing computer-based technology to deliver curriculum information and manage instruction. At the present time, efforts are being made to modify the system and expand it to other secondary vocational education centers and the technical and community college system within the state of Delaware.

Selected References

Conroy, William G. "Planning for Occupational Education with Misoe," Journal of Research and Development in Education. 7:91, Winter, 1974.

Cooley, William W. and Robert Glaser. "An Information and Management System for Individually Prescribed Instruction," in Richard C. Atkinson and H. A. Wilson, Ed., Computer-Assisted Instruction, New York, Academic Press, 1969.

Danford, Jan. "Is Your Career Development Program Tailored for the Real World," Journal of Technological Horizons in Education. 1:5, November, 1974.

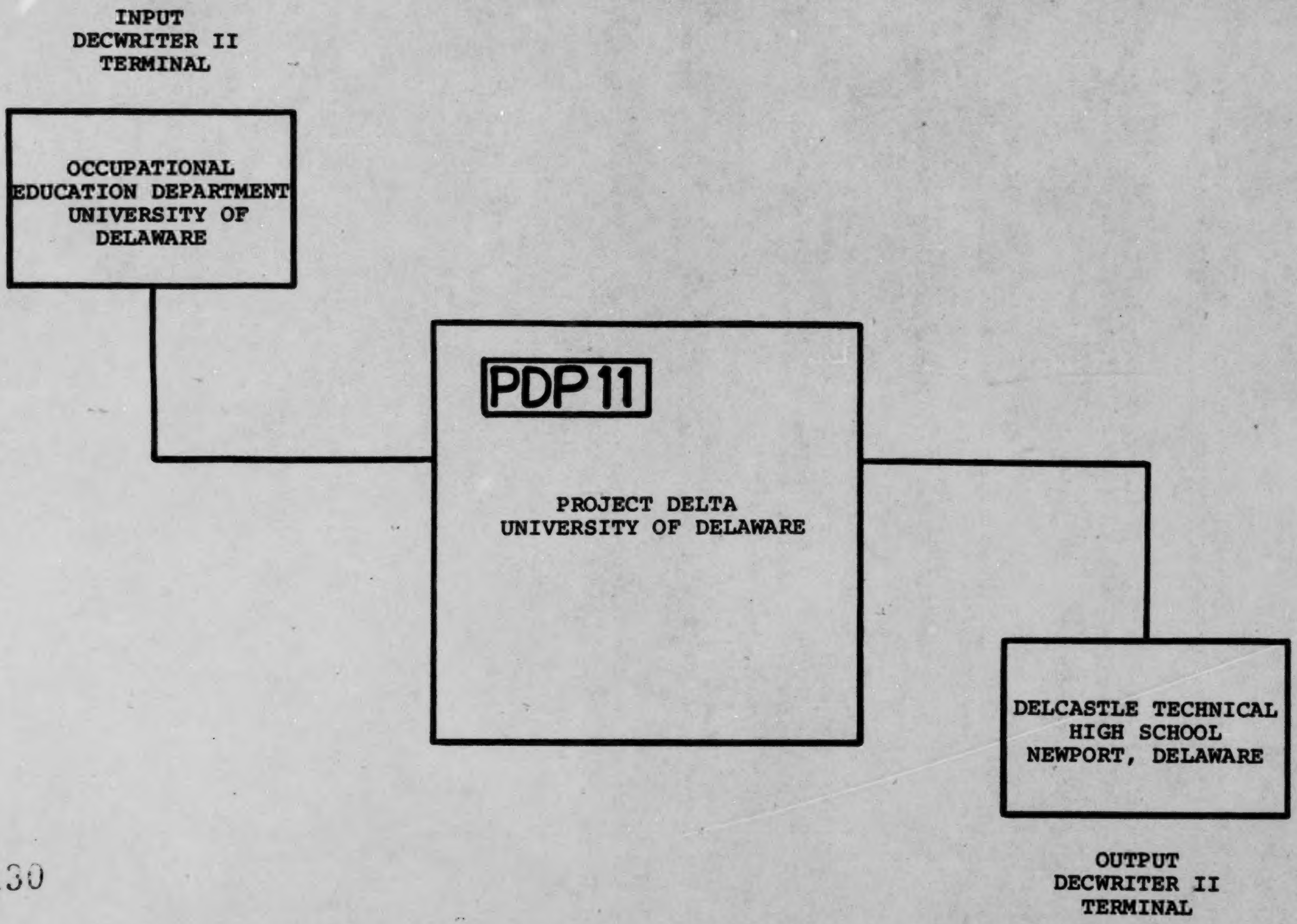
Dupuis, Victor L. and Paul E. Bell. "A Modular Approach to Vocational Curriculum," American Vocational Journal. 49:34, April, 1974.

Frantz, Nevin R., Jr. and Edward E. Boas, Jr. "A Computer Managed Instructional System for a Multi-Occupational Approach to Vocational Education," Presentation at the 14th National Conference of the National Society for Performance and Instruction. Atlanta, Georgia, April, 1976.

Frantz, Nevin R., Jr. Individualized Instructional Systems for Vocational and Technical Education: A Collection of Readings. Newark, Delaware: Vocational Instructional Systems, 1974.

Rannells, Bob. "Concept of the 70's -- Individualized Student Instruction," Agriculture Education. 8:1, July, 1973.

APPENDIX A


INPUT
DECWRITER II
TERMINAL
OCCUPATIONAL EDUCATION DEPARTMENT UNIVERSITY OF DELAWARE
PDP 11
PROJECT DELTA
UNIVERSITY OF DELAWARE
DELCASTLE TECHNICAL HIGH SCHOOL NEWPORT, DELAWARE
OUTPUT
DECWRITER II
TERMINAL

APPENDIX B

```
RUNOCINFO

OCINFO -- OCCUPATIONAL CURRICULUM INFORMATION
HELP IS AVAILABLE AT ANY TIME BY TYPING 'HELP'

WHICH CLUSTER: LIST

PE -- PERSONAL SERVICES
HS -- HEALTH SERVICES
C  -- CONSTRUCTION
M  -- MANUFACTURING
T  -- TRANSPORTATION
AB -- AGRI-BUSINESS AND NATURAL RES
PU -- PUBLIC SERVICE
E  -- ENVIRONMENT
HR -- HOSPITALITY AND RECREATION
FH -- FINE ARTS AND HUMANITIES
CM -- COMMUNICATIONS AND MEDIA
MD -- MARKETING AND DISTRIBUTION
MS -- MARINE SCIENCE
BO -- BUSINESS OFFICE
CH -- CONSUMER AND HOMEMAKING EDUC


WHICH CLUSTER: T

WHICH SUB-CLUSTER IN TRANSPORTATION: LIST

A  -- HIGHWAY TRANSPORTATION
B  -- RAIL TRANSPORTATION
C  -- AIRBORNE TRANSPORTATION
D  -- PIPELINE TRANSPORTATION
E  -- WATER TRANSPORTATION


WHICH SUB-CLUSTER IN TRANSPORTATION: A

WHICH SPECIFIC AREA IN HIGHWAY TRANSPORTATION: LIST

AM -- AUTO MECHANICS
SE -- SMALL ENGINES
AB -- AUTO BODY AND FENDER RPR
D  -- DRIVING
RM -- REPAIR AND MAINTENANCE
BH -- BAGGAGE HANDLING
DS -- DISPATCHING
SR -- SALES REPRESENTATION
SS -- SERVICE STATION OPERATION
DI -- DRIVING INSTRUCTION
CB -- COMMERCIAL BUS SERVICE
TS -- TAXI SERVICE


WHICH SPECIFIC AREA IN HIGHWAY TRANSPORTATION: AM
```

APPENDIX B

```
WHICH CLASSIFICATIONS: LIST

FOLLOWING IS A LIST OF ALL THE CLASSIFICATIONS.

          A.   COURSE OUTLINES.
          B.   CURRICULUM GUIDES.
          C.   INSTRUCTIONAL PACKAGES.
          D.   TASK INVENTORIES.
          E.   INSTRUCTIONAL OBJECTIVES.
          F.   INSTRUCTIONAL RESOURCES, E.G., FILMS, TAPES, ETC.
          G.   INSTRUCTIONAL ACTIVITIES,  E.G., GAMES, CASE STUDIES
          H.   EVALUATION DEVICES, E.G., PERFORMANCE TESTS,
               PAPER AND PENCIL TESTS.
          I.   FIELD TRIPS.
          J.   GUEST SPEAKERS.

WHICH CLASSIFICATIONS: I


CLASSIFICATION I (FIELD TRIPS)
------------------------------

1.)  T-A-AM  (I)

TRUCK DRIVING
     A CAREER RELATED FIELD TRIP CONCERNING THE TRUCKING INDUSTRY.
     THE PURPOSE OF THIS FIELD TRIP IS TO GIVE STUDENTS FIRST-HAND
     OBSVERATION OF THE SKILLS NEEDED FOR HANDLING A LARGE TRUCK.
     RECOMMENDED GRADE LEVELS: 11-12
     LENGTH OF VISIT: 1-1/2 HR.
     GROUP SIZE PREFERENCE: 20
     GENERAL LOCATION: NEWARK, DE.
     ADVANCE NOTICE NEEDED: 3 WEEKS

WHICH CLASSIFICATIONS: J


CLASSIFICATION J (GUEST SPEAKERS)
---------------------------------

1.)  T-A-AM  (J)

URBAN TRANSPORTATION
     A CAREER RELATED DISCUSSION CONCERNING ALL ASPECTS OF URBAN AND
     SUBURBAN BUS TRANSPORTATION.
     RECOMMENDED GRADE LEVELS: 6-12
     GROUP SIZE PREFERENCE: CLASS
     LENGTH OF PRESENTATION: ARRANGEMENT
     ADVANCE NOTICE NEEDED: 3 WEEKS
```

APPENDIX C

ENABLING OBJECTIVE # 2

OBJECTIVE

A. ANSWER CORRECTLY FOUR QUESTIONS ABOUT THE CLEANING AND INSPECTION OF THE CRANKCASE AND ACCESSORIES.

B. CLEAN AND INSPECT THE CRANKCASE AND ACCESSORIES ACCORDING TO MANUFACTURER'S SPECIFICATIONS.

RESOURCES

BOOKS: SMALL ENGINES, VOLUME 1.
MANUFACTURER'S REPAIR MANUALS

TOOLS AND MATERILAS: PAGE 25 IN SMALL ENGINES, VOLUME 1, HAS A LIST OF TOOLS AND MATERIALS NEEDED TO CLEAN AND INSPECT THE CRANKCASE.

ACTIVITY A

READ THE SECTION TITLED, 'CLEANING AND INSPECTING THE CRANKCASE AND ACCESSORIES,' AND STUDY FIGURES 17-21 ON PAGES 25-28 IN THE BOOK, SMALL ENGINES, VOLUME 1. IF NECESSARY, REFER TO THE REPAIR MANUAL FOR THE PARTICULAR ENGINE YOU ARE WORKING ON FOR FURTHER INFORMATION ON CLEANING THE ENGINES.

ANSWER CORRECTLY THE FOLLOWING QUESTIONS:

1. BECAUSE IT IS FLAMMABLE AND A FIRE HAZARD, __________ SHOULD NOT BE USED TO CLEAN AN ENGINE.

2. WHEN CLEANING THE ENGINE, IT SHOULD BE:
   A. HOT, B. COLD, C. WARM.

3. CHEMICAL DEGREASERS ARE FLUSHED OFF THE ENGINE WITH ___________.

4. A __________ SOLVENT SUCH AS MINERAL SPIRITS, KEROSENE OR DIESEL FUEL MAY BE USED TO CLEAN THE ENGINE.

CHECK

WHEN YOU HAVE FINISHED ANSWERING THE QUESTIONS, CHECK YOUR ANSWERS WITH THOSE LISTED BELOW. IF YOU HAVE MISSED ANY, GO BACK AND REVIEW THE RESOURCES UNTIL YOU CAN ANSWER THE QUESTIONS CORRECTLY. IF YOU ARE STILL HAVING PROBLEMS, ASK YOUR INSTRUCTOR FOR HELP.

1. GASOLINE
2. C. WARM
3. WATER
4. PETROLEUM

ACTIVITY B

CLEAN THE ENGINE BY FOLLOWING THE PROCEDURES ON PAGES 25 THROUGH 28 IN SMALL ENGINES, VOLUME 1.

REFER TO THE MANUFACTURER'S REPAIR MANUAL FOR THE PARTICULAR ENGINE YOU ARE WORKING ON IF NECESSARY.

CHECK

WHEN YOU HAVE FINISHED AVTIVITY B, HAVE THE INSTRUCTOR CHECK YOUR WORK. IF NECESSARY, REVIEW THE RESOURCES AND ACTIVITIES OR ASK YOUR INSTRUCTOR FOR HELP.

APPENDIX D

```
PROGRESS REPORT FOR 17-Mar-76 AT 12:06 PM

STUDENTS LISTED:
NUMBERS: 0001-0005

STATUS KEY:

0 = MODULE GIVEN WITHOUT PRE-CHECK
1 = PRE-CHECK GIVEN BUT NOT CHECKED YET
2 = FAILED ONE OR MORE EO'S OF PRE-CHECK
3 = PASSED PRE-CHECK (TPO COMPLETED)
4 = POST-CHECK GIVEN BUT NOT CHECKED YET
5 = PASSED POST-CHECK (PASSED TPO)
6 = FAILED POST-CHECK (FAILED TPO)

0001 EDWARD BOAS             T-SE-06 (5,09:05 PM,06-Mar-76,09:17 PM,06-Mar-76)
                             T-SE-06 (2,11:39 PM,06-Mar-76)
                             AG-OH-01 (1,12:11 PM,11-Mar-76)
                             T-AB-01 (6,06:18 PM,15-Mar-76,06:27 PM,15-Mar-76)
                             AG-OH-02 (6,07:04 PM,15-Mar-76,07:13 PM,15-Mar-76)
                             T-SE-02 (5,11:52 AM,17-Mar-76,12:03 PM,17-Mar-76)

0005 THOMAS PANTANO          T-SE-04 (5,09:11 AM,27-Jan-76,10:38 AM,29-Jan-76)
                             T-SE-01 (5,10:57 AM,29-Jan-76,09:57 AM,30-Jan-76)
                             T-SE-03 (6,10:24 AM,30-Jan-76,09:07 AM,05-Feb-76)
                             T-SE-02 (6,09:24 AM,05-Feb-76,10:07 AM,05-Feb-76)
                             T-SE-05 (6,10:17 AM,05-Feb-76,10:09 AM,06-Feb-76)
                             T-SE-01 (6,10:31 AM,06-Feb-76,09:29 AM,26-Feb-76)
                             T-SE-02 (4,09:27 AM,09-Feb-76)
                             T-AB-01 (1,09:53 AM,09-Feb-76)
                             T-SE-05 (6,09:21 AM,16-Feb-76,09:57 AM,24-Feb-76)
                             T-SE-03 (4,10:13 AM,19-Feb-76)

0006 DWAYNE PIERSON          T-SE-01 (5,09:31 AM,27-Jan-76,10:57 AM,27-Jan-76)
                             T-SE-04 (5,10:24 AM,29-Jan-76,09:46 AM,30-Jan-76)
                             T-SE-03 (6,10:10 AM,30-Jan-76,10:24 AM,03-Feb-76)
                             T-SE-02 (6,10:41 AM,05-Feb-76,10:18 AM,06-Feb-76)
                             T-SE-05 (6,10:43 AM,06-Feb-76,10:35 AM,19-Feb-76)
                             T-SE-05 (6,09:06 AM,09-Feb-76,09:39 AM,26-Feb-76)
                             T-SE-05 (6,09:10 AM,24-Feb-76,09:29 AM,02-Mar-76)
                             T-SE-05 (2,09:01 AM,02-Mar-76)

0007 GEORGE RIBYNSKI         T-SE-01 (4,09:23 AM,27-Jan-76)
                             T-SE-01 (0,09:35 AM,29-Jan-76)
                             T-SE-04 (6,08:59 AM,30-Jan-76,10:29 AM,19-Feb-76)
                             T-SE-03 (6,10:31 AM,30-Jan-76,09:14 AM,05-Feb-76)
                             T-SE-02 (5,09:17 AM,05-Feb-76,10:03 AM,05-Feb-76)
                             T-SE-05 (6,10:26 AM,05-Feb-76,10:00 AM,09-Feb-76)

0008 DOMINIC PALADINETTI     T-SE-04 (5,09:21 AM,29-Jan-76,08:51 AM,30-Jan-76)
                             T-SE-01 (5,09:21 AM,30-Jan-76,10:18 AM,30-Jan-76)
                             T-SE-05 (5,10:45 AM,30-Jan-76,09:40 AM,09-Feb-76)
                             T-SE-03 (6,10:03 AM,09-Feb-76,10:41 AM,10-Feb-76)
                             T-SE-02 (5,08:45 AM,12-Feb-76,09:04 AM,12-Feb-76)
                             T-SE-03 (1,10:22 AM,19-Feb-76)
```

AN INSTRUCTIONAL SYSTEM IN MATHEMATICS

Catherine E. Morgan
Director
Instructional Systems Development Project
Montgomery County Public Schools
Rockville, Maryland

Background

On October 27, 1975, the Montgomery County Board of Education adopted a policy on comprehensive planning for a learner-centered management support system--a system designed to enhance student learning through more effective instructional programs--a system designed to make clear the relationship among purposes, results, costs, and benefits--a system to provide meaningful information required for decision-making at all levels within the Montgomery County Public Schools.

The policy on comprehensive planning was the first step toward the goal which Superintendent Bernardo has called the main theme of his administration, that is, "to develop facilitative systems to provide maximum learning to students, and to support the professional staff at the most reasonable costs."

The second step toward this goal occurred on November 11, 1975, when the Board of Education adopted the Policy on Curriculum Development and Criterion-Referenced Testing. This policy authorized the Superintendent to begin systematic planning for the development and implementation of learner-centered instruction, the central element in a learner-centered management support system.

Directives included in this planning process were first, to establish a curriculum consisting of a set of specific objectives stated as student outcomes: to establish procedures for assessing individual student progress. These procedures would include criterion-referenced tests which enable the student and teacher to determine the extent to which the student has mastered essential objectives; to identify the supports required for teaching and learning--supports such as instructional guides, materials, and equipment; and, to establish procedures for collecting, analyzing, interpreting and reporting information on student progress and procedures for program evaluation.

Implementation of this policy was begun, and in January, 1976, the Board received the Superintendent's Five-Year Plan for Development and Implementation of Learner-Centered Instructional Programs in Essential Skills. This Five-Year Plan identified the instructional system in mathematics as the first to be designed, developed, pilot tested and implemented. By the fall of 1981, the plan calls for the implementation in all elementary, middle and junior high schools of instructional systems in mathematics, reading/language arts, science, and social studies.

The responsibility for the development of these systems was given to the Instructional Systems Development Project under my direction. This group also functions as the Division of Computer-Related Instruction and continues to perform those duties as well as the ones relating to the new task. Coalitions are formed with the subject coordinators and their staffs from the Department of Curriculum and Instruction. In addition, area personnel are also involved in the development and implementation.

The basic purpose for developing an instructional system is stated in the Montgomery County Goals of Education, which were adopted in February, 1973. Of the seven goals, the strongest statement is made for the Academic Skills: "The fundamental responsibility of the school is to develop programs that enable each child to acquire those skills basic to all learning. The achievement of all other goals depends upon the success of the school in ensuring that each student, according to his own abilities, attain the following skills: reading, composition, listening and speaking, mathematics, study, the arts, and observation."

As part of the Goals of Education statement, the school system made several commitments to the citizens of Montgomery County. In 1973 a commitment was made to develop and implement a comprehensive and flexible instructional program to achieve the goals and to provide the facilities, materials and equipment needed to enhance the effectiveness of this program.

A further commitment was made to continuous evaluation of the effectiveness of this instructional program and to regular reports on findings to students, parents, staff and the community. In addition, the school system is seeking new and different approaches when it is determined that they will contribute more effectively to the fulfillment of the goals. The learner-centered management support system as conceived for the Montgomery County Public Schools is designed to fulfill those commitments--to make quality education even better.

The Goals of Education document recognizes that the probability of success motivates students to learn and that the schools must create a program and maintain a climate in which every student has opportunities for success. The desire to increase the probability of student success motivates the development of instructional systems that will focus upon the learner. Simply stated, this system is a way of organizing to achieve the goals of the schools--assuring that each student makes continuous progress in acquiring basic skills and providing appropriate information regarding this student progress to all educational decision-makers including students, parents, teachers, administrators, and the Board of Education.

Within the context of a learner-centered management support system, each instructional system will consist of five parts: objectives, assessment, instruction, reporting, and management.

For several years the Montgomery County Board of Education has supported the development of specific instructional objectives and criterion-referenced tests. In the Department of Curriculum and Instruction, the mathematics curriculum, kindergarten through 8th grade, was being described as a set of objectives stated as student outcomes and organized for student and teacher use. This task involved input from classroom teachers, area teacher specialists and resource teachers. In the summer of 1975, the Board approved the present mathematics curriculum for K-8, a set of objectives called the Elementary and Pre-Algebra Mathematics Objectives. Mathematics was chosen as the area in which to begin development and pilot testing of an instructional system because of emphasis on mathematics in the Goals of Education document and because the curriculum had already been written and approved as a set of objectives. The mathematics curriculum stated as a set of objectives with assessment items provides a uniform, consistent curriculum on basic math skills across all Montgomery County elementary, middle and junior high schools.

The objectives have been divided into 26 categories such as numeration, geometry relations and statistical graphs. Of the 340 objectives, approximately 250 have been identified as key objectives. Each of these will be formally assessed by the system. Four equivalent sets of assessment items have been developed for each objective. All of the kindergarten objectives and many of those in first and second grades require the student to demonstrate understanding through the manipulation of concrete objectives. Most of the other assessment tasks can be in pencil and paper form or computerized. A few objectives can only be tested in paper and pencil form (i.e., bisecting angles, constructing line or bar graphs.)

Information on achievement is recorded for each student in a computer file. During the current school year, 12 elementary schools and 3 junior highs are involved in this developmental pilot. In 6 elementary schools, one in each administration area, students in grades 3-6 will use computer terminals for assessment. In grades K-2, assessment is conducted by school staff, and individual student results are entered in the student's computer file by a staff member using the terminal. These schools also use the computer-assisted instruction program in arithmetic. In the remaining elementary pilot schools, paper and pencil versions of the same assessment items are used on an individual or group basis. These items are graded and forwarded on a regular basis to the central office where the results are entered at terminals. In one junior high school, the students use terminals for assessment and in two additional ones, paper and pencil items are used.

In addition to the criteria of two elementary schools per area, one with computer-assisted instruction terminals, and of faculty commitment to the pilot, these 15 schools were selected to represent a cross section of Montgomery County schools and students. Factors considered were socio-economic background, school size, student achievement and aptitude, organizational pattern for mathematics instruction, and previous experience in the school with systematic instructional programs.

The system exists to help teachers teach children better. Instruction, including organization, methods, and materials, varies from school to school, even from classroom to classroom. The system has the flexibility to support various options for instruction. In the classroom, diagnosis and regular assessment allow the teacher to match methods and materials of instruction with student needs. One aim of the mathematics instructional system is to identify all materials and procedures which promote student learning.

To help the teacher, an instructional guide for mathematics was prepared. This guide includes a developmental sequence of objectives in each category and at least one sample teaching activity for each objective. Instructional materials, including approved textbook, manipulatives, games and filmstrips, have been coded to each objective. The guide will be continually updated, listing all materials found to be effective for teaching specific objectives.

A fourth feature of the planned system is the preparation and distribution of reports. Profiles and progress reports, both for individual students and for groups of students, are available to those who need this information prior to making decisions at the classroom, school, area, and county levels. For example, teachers receive current profiles showing mastery of objectives by individual student and by class. These profiles help the teacher plan instruction appropriate to the learning level of each student, to group students for instruction, to identify the relationship between instruction and student performance, and to note student progress.

During the pilot, users are being asked to try out and suggest improvements in these reports to make them as helpful as possible for instructional purposes and for the allocation of resources within the schools.

The fifth part of this instruction system in mathematics can be described as management. Included here is the essential task of training school staff to use the system and to aid in its development and in its continuous improvement.

Teachers in the 15 pilot schools received in-service training on each element of the system and how to use them in the classroom. Other tasks included orienting students to the system; setting targets for student performance; and analyzing each component of the system as it relates to student achievement and attitude and modifying components as necessary.

Teachers are receiving help in the form of the materials required to implement programs. These materials include clearly stated and agreed upon objectives; the assessment measures for these objectives; and the necessary instructional materials and supports. Teachers and principals are provided timely, useful information on student progress, information needed to plan instruction within the classroom and the school for maximum student achievement.

The student receives help in the form of individual diagnosis and assessment so that instruction can be tailored to his or her own needs. Students benefit by knowing what objectives they are to work on and by being involved in the management of their own instruction. Monitoring student progress helps students succeed and assures that teachers are given the supports required to provide effective instruction.

At the present time, the first major revisions are being contemplated based on information from student records and from formal feedback sessions with the pilot school staffs. Over 200 teachers are involved in the pilot and their suggestions are being incorporated into the system whenever possible.

The second instructional system, that of reading/language arts, is being designed at the present time and a portion of that system will be operating in a limited number of schools in the spring of 1978.

The MCPS systems are intended to help students experience success in learning; to help teachers provide effective instruction; to help administrators allocate resources based on needs; and to help all educational decision makers set priorities for the schools based on accurate and appropriate information on student achievement.

THE TEACHER AND TECHNOLOGY-THE WOODRIDGE STUDY OF CAI-A THREE YEAR STUDY

Katherine Hirschbuhl, Woodridge School District, Peninsula, Ohio

INTRODUCTION

Woodridge is not a large district; it has 1500 students. Located in a lovely rural area in the Cuyohoga River Valley between the two cities of Cleveland and Akron, it has a sharply contrasted socioeconomic scene. It is a mixture of large estates, small houses and increasing areas with trailer parks or apartments bringing to it a transient group that moves in and out each year.

In the spring of 1973, a Woodridge reading teacher and the district superintendent decided that help and special attention were needed for students experiencing failure and frustration while their peers were meeting with success. Funds were low and public support was difficult to obtain as bond issues and levies were being voted down in a repetitive pattern. They began to look for an answer. At that time, the school district employed two reading specialists, one at the elementary level which encompassed some 900 students and one part-time reading person in the senior high school. The teaching staff at the senior high school had pressed forward for a reading person after several years without one.

A needs assessment had been done at the 9th grade level through the California Reading Achievement Test. A 5-year study of English Usage Skills had been done at the 9-12 level by the Guidance Department. The Secondary Level Staff observed that revision of curriculum and heavy use of films was taking place. The overall analysis showed a decline in reading skills in the student body.

The situation was not common to the Woodridge School District, but it was a common occurrence nationwide. The transition from a print-oriented generation to a T.V.-oriented generation had taken place. Added to this tremendous change was the transition from tightly knit family structures to a high number of broken family structures, something once confined to the inner city schools as a problem. It was now common to the suburbs and rural areas as well. The entire school staff was concerned and in most cases self motivated toward doing something about what was taking place.

Sources of funding were studied as well as a starting point. It was decided to place the initial focus on the junior high level where there was a concentration of students funneling into the school from two elementary buildings. The teachers at that level had urged more emphasis on reading skills. An analysis of reading test scores by guidance, teaching staff, administration and several professors from The University of Akron further determined that the focus for remediation and promotion of reading skills would be planned for the junior high level (7th-8th grade) with future plans to extend the program as soon as possible to the intermediate level. What was being planned, in essence, was a reading team with learning centers to eventually cover all levels. With the establishment of the pilot program they hoped to work toward a cohesive and continuous program, K-12, aimed at reducing and eliminating feelings of failure and frustration in children with deficiencies in reading. A Title I grant was written by the reading teacher mentioned in the introduction, with the aid of the superintendent's and the assistant superintendent's offices. A sound testing program was planned so that the reading growth of each child taking part in the program could be objectively appraised.

From the beginning of the work on testing and planning, it was decided that no child would be screened out of the program due to low scores and testing histories which showed no potential for growth as it was too great a risk to take. The current research concerning I.Q. tests and Reading Tests had been proving that such eliminations were unwise educationally. There are children who do not respond well to tests, especially those with reading and learning deficiencies and sometimes these children turn out to be quite capable and even brilliant. The classic example in an extreme sense was Helen Keller who was labeled a mental defective because she could not be tested. After much one to one instruction, she was discovered to be a genius.

For the purposes of evaluation and screening, the California Reading Achievement Test, Level 4, was selected after a review of Burroughs and similar literature pertaining to tests. The CAT (California Reading Achievement Test) can be administered within an hour. Within the classroom, the reading specialists would use informal inventories and tests such as those designed by Bond, Clymer and Hoyt for individual analyses in individual skill areas. In addition to this testing, those students having learning problems with no apparent reading deficiency would be tested by a psychologist. Furthermore, those students in the Reading Program who showed a need for further diagnosis would be tested by the psychologist. All children in the Reading Program would be tested, also, by a Speech and Hearing Therapist.

One of the significant factors that the planners of the program kept in mind was the age level of the students entering the program. By the time a student in trouble reaches Junior High, the problem, if still there, has become compounded. Often the child has faced cumulative failures.

In order to chart reading growth, it was decided that the CAT (California Reading Achievement Test) would be given in the early fall and at the end of the school year, and analyzed by computer. All 7th and 8th graders (approximately 300) would be tested and all folders of testing histories would be screened. In addition to these procedures, parental and teacher recommendations would be a vital part of the screening process.

Within the design of the teaching and learning part of the program. varied instructional strategies and modes were established. These included a listening station, independent work areas, individualized work folders, Fun with Words corners, free reading stations, a tutoring station and a place for small group classroom sessions.

In a large old classroom next to the boiler room, in a 40 year old building, a reading center was established with some help from Title I. A coat of paint, some bright furniture, and dozens of books were contributed and bought. Some old carpeting was donated, cut up and placed in the large closet and in sections of the classroom. A box of corrugated wall covering was rummaged out of supplies and the closet walls were lined with it. The closet door was inserted with plexiglass with holes drilled into it. A computer terminal was placed in the closet. A structured program of skills was incorporated along with a Hooked on Books Approach, ie. "Books! Books! everywhere and lots of things to read." There was a tremendous excitement in the old building and the students and the staff built something where nothing had been -- and all of them used their power and the power of technology to prepare for the great age ahead -- the age of television, computers, satellites and more. Twenty five years ago the computer had just come into focus. Twenty five years hence was the dawning of a new century.

The Computer assisted instruction station was to be used as a supplemental learning tool. Because of its self pacing, individualized program of skills with drill and practice, tutorial lessons and constant responses to students, it provided both diagnostic and prescriptive instruction to 35 students on a regular daily basis. Each lesson was 10 minutes in duration. A child in the reading program received a 30 minute small group lesson or an independent lesson with a reading specialist as well as the 10 minute CAI lesson on a daily basis. The reading classes were scheduled out of the Language Arts classes so as not to place additional class time on students already needing extra time and help. The CAI lessons were scheduled through study halls if possible. The children were urged to spend their study hall in the Reading Center getting help and guidance for their content area subjects. Because of these procedures, many children came to the Reading Center several times a day and a large group opted to spend lunch periods in there as well. A part time aide was hired that first year to help keep the center's schedule smoothly flowing and to help the reading specialist put out reports to the staff and build materials for the center as well as for various content area subjects.

The Reading Center that first year was so busy that a reporter following the reading specialist around for a few days observing the enthusiasm of the students noted that from the minute the teacher's car pulled into the parking lot she was greeted by two boys who rushed to her car to get the key to the room and to the CAI station where they quickly started up the terminal and that the terminal went all day continuously with few vacant moments at that station. Kids followed the teacher everywhere asking questions about the CAI. Some even knocked on the Ladies' Room door asking for her and inquiring as to when they would get a chance to play with Herbie, their name for the terminal. The name had been decided by student vote with Walt Disney's Love Bug in mind.

The Reading program was covered by several newspapers several times and the center of attention was Herbie. One of the unexpected occurrences in the program was the amount of protest from students not in the program or using the CAI. Instead of being viewed as a place for the underpriviledged as many remedial programs are, some individuals felt underpriviledged to be denied participation.

SCHOOL YEAR 1973-74

At the end of the first year (1973-74) evaluation showed that seventy nine students had been actively enrolled in the program, some for 1/2 of a year and others for the entire school year. The average growth was 2 months for every month in the program. The CAI had not been initiated into the curriculum until January of that year. Those students receiving CAI lessons as an added enrichment did better as a group than those who did not receive it. Sixty nine of the seventy nine students were 2-1/2 years or more below grade level. Some were reading at primary level. Ten of the 79 students had emotional problems seriously affecting their school life and were, therefore, included in the program. For example, one student's parents had serious marital problems for several years. The year before he came to us, it was discovered that his father had a terminal disease and the father passed away during that 1973-74 school year. The boy spent every free moment in the Reading Center. His report cards showed all F's for months. His great interest was the CAI to the point that he spent many weekends at the University of Akron's CAI Center learning Coursewriter and APL. This type of student was placed in the program to benefit from the increased attention, individualization, and new horizons that could be offered. There were students in our program whose parents had moved seven times in seven years.

SCHOOL YEAR 1974-75

The following year (1974-75) had CAI for the full year as an active part of the program. Seventy one students were in our program. Forty seven were placed on CAI lessons according to seriousness of need.

> Note: Ninety Five percent of the students in the program receive CAI lessons. If they do not receive them in 7th grade they receive them in 8th grade -- even if they are no longer in the program. The only students not receiving CAI are those who express negative feelings. We do not make it a stringent requirement.

The 1974-75 data showed 2 months growth for every month in the program. An interesting observation of data was the class mean for the two grades. At Woodridge there is a historical pattern of falling right on the mean as a group. Ex. Seventh grade 7.0 - 7.1 Eighth Grade 8.0 - 8.1. This is due to the fact that our data has shown for years that we have a large group high above the mean and a large group scattered below the mean and a tiny group around the mean. In essence, the two groups, upper and lower, balance each other out to an average that falls within the mean. In 1974-75, however, both grades were 5 to 6 months above that mean. Furthermore, the same difference was noted between the CAI group and non-CAI group in Reading. One has to be careful in looking at the data as the students on CAI were the most severely retarded readers and although the non-CAI group had higher scores, the CAI group had more growth. The CAI group showed 3 month's growth for every month in the program.

During the 1974-75 year, there was a full time aide as well as full time CAI in the program.

1975-76 SCHOOL YEAR

Because of the success of the Junior High Program, we set out to build a similar program at the intermediate level. The support of the parents and the board was overwhelmingly good. The reading specialist at that level was given concentration at the Primary Level and another reading specialist was placed at the Intermediate Level. We were now covering K-12.

The CAT testing program was extended to cover 5th, 6th, 7th, 8th grade. Again, CAI was not established until January in the first year of the new intermediate program. The pre-test and post-test means for the student population tested were as follows. (All students in those grades were tested.)

TESTING PICTURE

California Reading Achievement Test

Means -- where our students average out as a group on a standardized test given nationwide.

	Vocabulary	Comprehension	Total Reading Ability
8th Grade	8.3-9.4	8.1-8.7	8.2-9.0
7th Grade	7.1-8.9	7.1-7.7	7.1-8.3
6th Grade	6.6-7.0	5.5-6.6	6.3-6.8
5th Grade	5.3-6.4	4.7-5.9	5.1 6.2

Seventy one students were covered by the Junior High Program and forty four students at the Intermediate Level. (Note: Seventeen students from 3rd and 4th grade were channeled into the Intermediate Program but did not receive CAI. They were in another building.) The average growth was 1-1/2 month's reading growth for every month in the program. Again the CAI student group showed more gain in individual growth spans. They showed 2 month's growth for every month in the program. Of the seventy one students in the junior high, sixty five students were provided individualized lessons with drills on basic skills. Six of the sixty five students went to the University of Akron for lessons at the CAI Lab. Furthermore, the CAI schedule began earlier with students going on CAI at

7:45 until 4:00 with a few parents bringing youngsters in for evening lessons.

The variables in the 1975-76 year were:

1. The program had expanded into several buildings and part of it was new. (Intermediate.)
2. The Reading Center was flooded with requests from the entire staff for help in building content area reading materials.
3. Students from throughout the school came to the Reading Center so that there was a continuous stream of students into the center at all times, in addition to the regularly enrolled students in need of help (Junior High.) And many students wanted to experiment with the CAI.
4. The intermediate program had a seasoned teacher who was in his first year as a reading specialist -- building a new program in two buildings. (Intermediate.)
5. The English Program was involved in projects and students did not come on as regular a basis as in the past. (Junior High.)

THE READING AND CAI PROGRAM'S IMPACT

The Woodridge schools have experienced and observed the following:

1. The Director of Guidance whose office is located at the high school has seen a marked improvement in the incoming freshmen, particularly in self image, discipline, and attitude.
2. Dozens of letters have been written by parents praising the Reading Program and offering support.
3. The Language Arts Department at the Junior High School has been actively involved in supporting the Reading Program and helping to build it.
4. Newspapers and T.V. stations have covered the program.
5. Visitors from surrounding school districts, other states and other countries have found their way to the tiny Woodridge School district.
6. An awareness of Computer Literacy has developed and field trips are made to Computer Centers and The University of Akron's CAI Center.

7. A group of youngsters not enrolled in the program have developed computer skills, learned computer languages and have displayed hidden talents,

8. A strong association between professors, teachers and students has developed through the University and School CAI alliance,

9. A CAI Math program has recently been developed and is about to be initiated at Woodridge Junior High.

10. Teachers have strongly supported something they feel is needed and have given extra time and study freely.

CONCLUSIONS

1. CAI is an effective instructional tool which can be effectively used in remediation of skills in the classroom.

2. CAI can serve as a learning enrichment tool as well as a tutorial learning tool.

3. Students who were actually suffering learning loss can improve to above average gains in learning if concentration and attention is provided.

4. CAI helps to erase the stigma attached to special reading programs. Both advanced and remedial students alike flock to the Reading Center, working together on their studies and asking to be placed on CAI.

5. An Open Door Policy between superintendents and teachers can lead to building very worthwhile instructional programs for children.

6. As other studies have pointed out and as Title I has stressed, the greater the coverage of students by one teacher, the less growth for those who are seriously deficient in basic skills, ie. Small groups are beneficial to the learning process.

7. As other studies have shown, students who have not learned through traditional methods will benefit from changed methods and new strategies,

8. Students respond better to regularity in conditions, ie, daily sessions, meetings, knowing what is expected, especially students in need of help. (This, too, is supported by studies.)

9. A good reading program is helped by the cooperation of the entire staff, ie. reinformcement through the content area subjects.

10. Parents and Board Members respond well to carefully collected data and open observation of what is taking place in their school district.

11, It is possible for Universities and schools to work closely together on new programs for the benefit of the future advancement of the educational process.

HOW CAN A TEACHER FIND OUT ABOUT COMPUTERS AND CAI?

There are huge amounts of literature on the subject. Universities often have CAI centers with schools hooked up to them. The University of Akron has a workshop for teachers on this subject. There are schools with CAI where teachers can observe it in action.

COST EFFECTIVENESS

An identification of what cost effectiveness is concerns what factors should be considered in examining cost/benefit ratios under the assumption that amount learned and students saved are real cash value items.

Some schools have their own computers and are, therefore, automatically set up for CAI. The school without a computer can be easily hooked up to the nearest university computer center and CAI network via the telephone. There are Indian reservations hooked up to universities in California. What does it cost at present? It depends on what your needs are.

Our terminal goes from 8:00 to 3:15 every day without stopping. A few parents come in the evening to take advantage of its unused richness. Think about the last time you visited your doctor and filled your prescription. What was the cost? Think about what it takes to keep one man in prison for one year -- over $12,000.00. Someday and somehow our values will begin changing and the GNP won't show cosmetics, booze, and tobacco as more important than education.

What it adds up to really is the cost of a new car. This, however, serves many and not a few on a one to one basis.

My terminal costs me $250.00 a month with no summer rent. Some cathode-ray tubes cost more; they require a separate line. The more terminals in one place, the less expensive the charge per terminal. I have an aide. The cost for CAI comes to approximately $3,000 for each CAI set up (one terminal). This includes maintenance, phone costs and hook-up. The effectiveness has been seen in student scores, in student enthusiasm, in numbers of students covered.

THE FLEXIBILITY OF CAI IN THE CLASSROOM

CAI is not locked into a few skill areas unless a school district selects that option. When a terminal is hooked up to a computer, all of the programs on that computer are available to the students or teachers using that terminal. A fourteen year old student in Peninsula, Ohio took a freshman college course in Chemistry, along with a computer language, Fortran. Another thirteen year old took Probability. Others are taking a very popular course in grammatical lessons, Basic English. Many are taking reading skills at various grade levels in such areas as: Vocabulary, Literal Comprehension, Interpretive Comprehension and Word Analysis. There are games on the computer which are enjoyable and beneficial at the same time. And just for amusement, a few clever souls have placed golf and tennis on the cathode-ray type terminals. The future prospects of what can be done with it are innumerable and myriad in nature. For the few who call it a frill or a toy, I ask them to look at history, to look around themselves, to become a part of this great age of television, satellites, and computers. Books and people will never become obsolete but ostriches in the midst of a great technological advance will become obsolete. That is the only way the artist in every person can stay alive and that we can make technology serve us and enrich us. We are actually cheating children by not introducing them to something that is now an integral part of our society and environment.

NORTHEAST OHIO CAI NETWORK

The Woodridge pilot project has developed into a growing CAI Network between public schools and the University of Akron. At this time a dozen terminals have been set up by various public schools in Northeast Ohio and are being used in related ways to the Woodridge CAI Instructional Model. Several college and universities are a part of the network. A CAI Conference for Superintendents in Northeast Ohio was held at The University of Akron one year after the project had begun. Teachers and administrators have suggested variations of the Woodridge model. Many local teachers, administrators and professors enrolled in a CAI Workshop at the University of Akron this summer. The future of CAI in the classroom looks promising.

SUMMARY

Computers in schools are becoming a reality. Gigantic memories can now use laser beams to store vast amounts of information which is easily accessible. Using a terminal as one of five or six instructional stations, along with small group teaching sessions and one to one teaching situations while building a hooked on books approach has brought me into one of the most exciting and rewarding teaching experiences in my 15 years of classroom duty. Children follow me everywhere and they meet me in the parking lot in the morning. They make visits on their own to computer centers, and most of all, they are reading and spelling better. The amazing thing to observe, however, is in their behavior. Many become more serious students and develop better self images. I have more time to spend with them, more time to listen, more time to teach them the extra things. Furthermore, the reading room is thought of as a progressive and intellectual type of place and a fun place -- not a place for dummies. All of this has made the extra work, the extra investment worthwhile. The University of Akron says that our beginning terminal, that one little terminal, was used more than any terminal in their network. There have been times that I have worked overtime, but life is not a lock step schedule.

FACULTY AND MODULE EVALUATION VIA PLATO

Arthur D. King*
Stephen J. Summers
Marie E. Shanley

Florida's College of Dentistry has a highly self-instructional undergraduate dental curriculum that permits unique student completion times for each module. Each module involves team teaching. The curriculum is based on a systems model of education which necessitates revision and improvement. Because of the self pacing of students and the varying number of faculty teaching a course or module, the collection, analyses and formatting of students' reactions to the faculty's teaching performance or the modular instructional activities presents some logistical problems.

This presentation will describe why we used PLATO to assist us with faculty/module evaluations and our experiences with the faculty, students and administration.

Approach to Faculty/Module Evaluation

As with most colleges, the performance expectations of faculty members fall primarily in the areas of teaching, research, and service. In the College of Dentistry, teaching includes academic administration and the development of self-instruction teaching packages as well as classroom teaching, laboratory and direct clinical contact with students. Service includes care to patients through intramural faculty practice as well as service to the community, university and college. Our major concern in this presentation is with teaching effectiveness. We feel teaching effectiveness is best evaluated by students although peer evaluation is also important.

The main purpose of our faculty evaluation programs are to: 1) guide faculty development for improved performance in affective, cognitive, and performance aspects of the teaching process and 2) provide data for merit, tenure, and promotion considerations.

The main purpose of our module evaluation is to collect student opinions about the quality of each module/course included in the undergraduate program. The results will be used by the individual faculty and the departmental chairpersons responsible for the offering of the module. The Curriculum Committee as well as the Department of Dental Education will also use the results to guide revision. The role of evaluation in bringing about change is widely recognized and accepted. We feel our changes should be implemented in a systematic and thoughtful manner. To do this, we must continually assess the difference between actual performance and desired performance. Our educational philosophy employed is based on a systems approach and implies a "feedback" loop for improving the instructional process. Faculty need "feedback" to improve their teaching performances in their modules. We have used PLATO to assist in the administration, analyses, and formatting of faculty and module evaluation results.

---

* Dr. King is Associate Professor in the Department of Dental Education and Coordinator of Computer-Based Instruction for the Learning Resources Center. Mr. Summers and Ms. Shanley are Programmers and also manage the PLATO Lab.

Why PLATO?

We have a four-terminal PLATO system connected to the Florida State system in Tallahassee. PLATO capabilities have been investigated for applications in the dental curriculum since January of 1975. We wanted faculty and module evaluation conducted. We were experimenting with PLATO to monitor student progress in the curriculum. Module progress reporting requires a student to use a PLATO terminal to enter enrollment dates and completion dates when he/she has started or finished a module. PLATO fulfills the design needs of the faculty/module evaluation activities by permitting the presentation of relevant questions at the appropriate time with minimal effort required from the faculty and students. Using a PLATO terminal to have a student select an A through F for each question seems easier than using a questionnaire with an optical-scoring answer sheet that requires the coordination of the questions and answers, making of the marks, organizing the results according to faculty/module, and analyzing the results.

PLATO is available from 8:30 a.m. to midnight each day so that it is usually available when a student finishes a module. The analyses and formatting of results is conducted as each student completes each evaluation. A PLATO terminal cannot hear, see, or talk to the students completing the evaluation and should minimize the influence of an instructor whose presence might influence student decisions.

Since PLATO was already available, the use of it to handle another applicaion would permit additional rationale for cost justification. PLATO costs were being distributed to CAI, CMI, and student progress applications so if we added another application that also handles a sometimes sticky issue, i.e. faculty evaluation, then the administrative support of the system should increase.

We have been using PLATO since the spring of 1976 for the administration, analysis, and formatting of faculty/module evaluation activities. To summarize why we selected PLATO, it was because of the potential conveniences offered to the student, the measurement specialist, and the faculty. Students were already using PLATO and by incorporating faculty/module evaluation with existing PLATO applications, we only have to maintain one system.

Method

Subjects:

There were 180 undergraduate dental students requested to rate 35 modules and there were 95 faculty assignments in the modules. The evaluations were to be completed between December 1, 1976 and January 7, 1977.

Procedures:

The Department of Dental Education designed and edited questions that could be used for improvement of teaching, improvement in the modules, and for appraisal of teacher performance. The faculty evaluation questions focused on the functions contained in an analysis of teaching activities that was completed during 1972-1974. The module evaluation questions focused on hypotheses that have been stated concerning the modular curriculum. For example, the modular curriculum should have clearly stated objectives; the curriculum permits self-pacing, the evaluation instruments are objective, valid, and reliable. The dental education faculty (3) examined several existing questionnaires but many of the questions were selected from Lawson's book (1974) on formative evaluation. Lawson's book has a unique statement that reads...."instruments contained herein may be duplicated by the purchaser for non-commercial, education related utilization."

The modified set of faculty evaluation questions with instructions were submitted to the chairpersons and deans for review and approval of how they would like to have students evaluate faculty. After two meetings with this group, the procedure was approved to have each student complete the faculty evaluation procedures as each module is completed (student has taken the posttest and may or may not know results). The questions were also reviewed and changes made by six selected students. Each question was stated in a positive manner and each question required the student to select one of the following choices for each question:

a.	outstanding	(1)	a.	strongly agree	(1)
b.	good	(2)	b.	agree	(2)
c.	average	(3)	c.	neutral	(3)
d.	poor	(4)	d.	disagree	(4)
e.	dreadful	(5)	e.	disagree strongly	(5)
f.	no comment/not applicable		f.	no comment/ not applicable	

The module evaluation questions were developed in a similar manner but the Curriculum Committee was the advising group as well as the students.

During the spring and summer of 1976, the students participated on a voluntary basis. We found that the students would not participate (only 40%) and that a few faculty would have 40-50 student ratings and other faculty none. In November of 1976, the college's administration made it mandatory for students to rate the faculty. If a student refuses to rate faculty, he/she meets with the faculty evaluation coordinator to explain his/her rationale for not participating.

To obtain student cooperation, the faculty evaluation coordinator met with each class and requested their participation, explained the PLATO procedures, reassured their confidentiality. Memos were posted in the Instructional Materials Center, in the clinics, near the student mail boxes, that contained an explanation of procedures. Also, the number of students completed and the number needed to complete faculty evaluation was posted daily in the IMC. Also during the holiday break, 78 students were sent notes reminding them to complete their evaluation responsibilities by January 7, 1977.

Although our long-range plans are to have each student rate each module and faculty teaching team as he/she finishes a module during the fall data collection period, we had the students rate several (3-8) modules and several (6-24) faculty at one session depending on the student's activities during the 1976 summer and fall quarters. The students were encouraged to make one-hour appointments to complete their faculty evaluation. They were also encouraged to complete them while the faculty evaluation coordinator or the programmers were present to provide assistance and to observe and handle problems encountered by the students.

The following provides a narrative documentation of the two PLATO lessons concerning faculty (fee) and module (mode) evaluation:

Faculty Evaluation - Documentation

Purpose: To collect data on the evaluation of dentistry faculty and to provide a display of results for each faculty member with a comparison of means and standard deviations between the individual faculty member and the faculty as a whole on each question.

Storage Considerations:

1) Number of Faculty Members - currently 71
2) Number of Questions - currently 27
3) Number of Choices/Question, or Scale - 1 to 5
   (1) on scale meaning very little improvement needed
   (5) on scale meaning extensive improvement needed
4) Counter for each possible choice to provide raw data on each question.
5) In order to provide a rapid display of information without having to calculate the standard deviation for each line of results, it was decided to keep the required data to calculate the standard deviation. These would be the number of observations, sum of the observations, sum of the squares of observations and the current calculated standard deviation itself.
6) The means could be calculated quickly when providing the number of observations and the sum of the observations by a simple division.
7) An upper limit on the numerical values for all variables had to be estimated to determine the segment size needed. Segmenting of variables was done whenever possible to reduce storage requirements.
8) Three student variables were segmented to provide a means of telling the student which faculty he had evaluated or who needed evaluation based on which module was previously done.

Calculations:

1) The number of observations under each choice was by question number for each faculty member.
2) The mean was determined by the following formula:

$$\text{Mean} = \frac{\Sigma x}{N}$$

Σx = sum of observations
N = number of observations

3) The standard deviation was determined by the following formula:

$$\text{S.D.} = \left[\frac{\Sigma(x)^2 - ((\Sigma x)^2 \div N)}{(N-1)} \right]^{0.5}$$

where S.D. = standard deviation
Σx = sum of observations
Σx^2 = sum of squares of observations
N = number of observations

Module Evaluation - Documentation

Purpose: To collect data on the evaluation of dentistry modules and to provide a display of results for each module with a comparison of means and standard deviations between a module and across all modules for each question.

Storage Considerations:

1) Number of Modules - currently 70
2) Number of Questions/Module - currently 22
3) Number of Choices/Question, or Scale - 1 → 5
   (1) = meaning very little improvement needed
   (5) = meaning extensive improvement needed
4) Counters for each possible choice to provide raw data on each question
5) In order to provide a rapid display of results without having to calculate the standard deviation for each question across all modules and each module, it was decided to keep the required data to calculate the standard deviation. These would include the number of observations, sum of the observations, sum of the square of the observations and the current calculated standard deviation itself.
6) The mean could be calculated quickly when providing the number of observations and the sum of the observations by a simple division.
7) An upper limit on the numerical values for all variables had to be estimated to determine the segment size needed. Segmenting of variables was done whenever possible to reduce storage requirements.
8) Two student variables were segmented to provide a means of flagging which modules had been evaluated and to link to the faculty evaluation lesson to tell the student which faculty were on the teaching team of the module evaluated.

Calculations:

1) The number of observations under each choice was kept by question number for each module.
2) The mean was determined by the following formula:

$$\text{Mean} = \frac{\Sigma x}{N}$$

Σx = sum of observations
N = number of observations

3) The standard deviation was determined by the following formula:

$$\text{S.D.} = \left[\frac{\Sigma(x^2) - ((\Sigma x)^2 \div N)}{(N-1)}\right]^{0.5}$$

where S.D. = standard deviation from mean
Σx = sum of observations
Σx^2 = sum of squares of observations
N = number of observations

Use of Faculty Evaluation Data

The student evaluation of faculty is reported annually or after a request by a chairperson, dean or individual faculty member. The Department of Dental Education provides a written report which contains the individual results. The number of students who have evaluated the faculty is indicated as well as scores with standard deviations for each of the 27 questions. For comparison purposes, the means and standard deviations for all faculty are included for each question. Students do not have access to faculty evaluation results. The faculty evaluation results are sent with a cover memo explaining the data collection procedures and suggestions on how the results may be used. If there are unusually low ratings, the faculty from the Department of Dental Education meet with the individual teacher to discuss and interpret the results and offer ideas and methods for improvement. Faculty acceptance and use of these results is a subtle and complex process. Individual arrangements are made between the faculty, Department of Dental Education, and the Learning Resources Center to promote the development of teaching skills that he/she feels need improving.

The dean, respective departmental chairperson, respective faculty member, and the coordinator have access to the evaluation data. These individuals interpret the results of the data. Appendix A is an example of how the results are provided. During the fall administration of faculty/module evaluation, the results were manually copied from the terminal. We are currently attempting connection of a DECwriter terminal with an interface provided by Mr. Fred Ebeling of the University of Arizona. The University of Illinois has sent tapes of the lessons "typer" and "typer data" to assist us in obtaining our "hard copy" capability.

Results

As of January 7, 1977, 148 of 180 students completed their faculty/module evaluation activities. The students who participated were cooperative and did not become upset despite spending as long as 1 1/2 hours rating the faculty and modules. The average time needed by the students to rate the faculty and modules was approximately 45 minutes. The results indicate that the questions were sensitive to make discriminations among faculty and modules. The data has been distributed but the faculty have not had time to react to the results. The faculty evaluation procedures has generated enough faculty interest to require the coordinator's explanation at the fall faculty assembly. The faculty assembly's steering committee has been appointed to review the faculty evaluation procedures. The fall administrations' results will be used to show the advantages and disadvantages to our PLATO-based faculty evaluation system.

Discussion

A major weakness in our program prior to the fall administration has been inadequate feedback from a large enough percentage of the student body for each faculty to feel secure that the evaluations by students were valid. For that reason, we implemented a mandatory system whereby as each student completes a module (course), he/she must either evaluate the faculty members involved in those activities (including clinical activities) or submit to the coordinator of the evaluation program a justifiable written statement as to why he/she cannot evaluate that particular faculty member at that particular time. Students may not progress to the next module in that sequence unless they have completed the evaluation component of the previous module.

A strong recommendation is that each institution, with significant faculty input, design their own programs for self, student, peer, and higher level administrator evaluation. Schools have different goals and different approaches as well as different requirements for their faculty members. It may be possible to provide an ongoing, comprehensive, broadly-based evaluation system on a manual basis, but it would appear to us that a computer-based program permits data collection at the appropriate time with efficient use of student time. The evaluation system must be sensitive to detect individual teaching differences but also be simple so that students can complete the evaluation in a short period of time. From the limited experience we have had, it would appear that frequent and ongoing evaluation as conducted on PLATO, especially that from students, is more convenient and permits more comprehensive evaluations than those done once or twice a year.

Berquist and Phillips (1975) suggest the following steps to improve the quality of student evaluation procedures: 1) use factor analysis to reduce the number of items; 2) the question should be constructed to be minimally evaluative and maximally descriptive to encourage instructional improvement. On the distribution of results, Berquist and Phillips suggest faculty evaluation data (that involves tenure, salary, and promotion decisions) should be made available to individual faculty, chairpersons, and deans and the evaluation data directing faculty improvement should be made available to the instructor.

The faculty member is the most critical factor in any effective program designed to improve instruction. But as Welbeck (1976) has reported there is a triad of components that relate to improving instruction. The triad includes faculty development, student development, and software (instructional materials) development. Each of these components need some system of evaluation to determine who, what, when, where, and how they are developing and in what direction. Because students are a major source of data on faculty, the more knowledgeable or "developed" the student is concerning the teaching/learning process, the more cooperative and informative the student learner will be in evaluating the faculty. The students must understand how the results are used and observe the effects or their willingness to provide the data diminishes. Students need to encourage faculty to improve their instruction. We have found the student evaluations conducted via PLATO have created a new medium for student-faculty interaction. As we strive to improve our college, we encourage active student participation in decisions concerning instruction by serving in a helping role to provide faculty information on his/her performance. The essence of Welbeck's suggestion is to be comprehensive in considering faculty development because the learner and the instructional materials (software) will also need to be developed to permit faculty to fully develop their entire teaching/learning potential.

Faculty evaluation should be keyed to faculty development or you risk the loss of the faculty's confidence in their teaching. Consider how you might feel after receiving poor ratings and there were no or very, very few resources available to permit improvement. Our guess is frustration. Faculty evaluation procedures must be sensitive to determine strength and weakness in the faculty's performance (diagnosis) but there must be a viable teaching prescription (treatment) for improving the weak areas. Evaluation without opportunities for training becomes a threat rather than an instrument for change.

Faculty acceptance is most critical. Faculty acceptance is not encouraged by the use of insensitive, often arbitrary, reliance on student generated evaluative ratings. The more that the faculty request the evaluation (rather than forced); the more that is descriptive (diagnostic rather than evaluative); the more concrete (rather than general); and the more trust it generates the greater the chance for a successful faculty evaluation/development program.

Despite the potential for faculty being negative toward PLATO technology because PLATO could be viewed as means of documenting their performance, it has not been the case. PLATO is used in this application as a convenient man/machine interface that questions students about the module instruction and faculty teaching.

The disadvantages we have identified in our applications are as follows:

1. One issue involves the confidentiality of the data base. Although we do not permit students to examine individual faculty results nor do we allow faculty to examine another faculty member's results, we are continually looking for methods to increase the security of the data base.
2. The lack of convenient "hard copy" from the student's input and results is a serious disadvantage. By not having the individual student's input stored provides confidentiality for the student but eliminates the possibility of using the data for statistical analyses. We currently keep results by questions but not by students.
3. Another critical issue is to insure that the student rates a faculty and module when he/she has finished a module, but not permit the student to repeatedly rate the same module/faculty.

We are encouraged by our current evaluation system but recognize that it must be continually improved. Our PLATO-based evaluation program encourages student participation and permits better coordination of effort within the college. We plan to use the module evalution results to guide curriculum revision. We are able to concentrate on curriculum improvements for three departments each year with the cycle repeating itself every fourth year.

REFERENCES

Berquist, William H. and Phillips, Steven R, A Handbook for Faculty Development, The Council for Advancement of Small Colleges, Suite 750, 1 Dupont Circle, Washington, D.C. 20036, June, 1975.

Welbeck, Paa-Bekoe, The Triad of Instructional Development, occasional papers published by the Division for Instructional Development, Association of Educational Communications and Technology, pp. 1-9, August, 1976.

Lawson, Tom E., Formative Instructional Product Evaluation: Instruments and Strategies, Educational Technology Publications, Englewood Cliffs, New Jersey 07632.

Copies of the module evaluation and faculty evaluation questions may be obtained by contacting Art King at College of Dentistry, University of Florida, Box J-406 JHMHC, Gainesville, Florida 32610 (904-392-2660)

Module Evaluation By Students

University of Florida/College of Dentistry

Module Number: 999 Module Title: Good Tooth Date: 1 /10/ 77

Number of Students Rating: 16

Question Area	Mean$_1$	SD$_1$	Mean$_2$	SD$_2$
1. quality of module	1.1	0.3	2.4	1.0
2. clarity of information	1.3	0.4	2.5	1.0
3. goals and objectives matched	1.1	0.3	2.2	1.0
4. objectives and posttests matched	1.2	0.4	2.5	1.2
5. faculty student interaction	1.1	0.3	2.3	1.1
6. efficient scheduling	1.1	0.3	2.3	1.1
7. interactive seminars	1.2	0.4	2.3	1.1
8. effectiveness of instruction	1.3	0.4	2.4	1.1
9. motivational students	1.5	0.6	2.5	1.1
10. quality of objectives	1.6	0.7	2.5	1.0
11. relevance of module	1.1	0.3	1.8	0.9
12. quality of media	1.5	0.6	2.5	1.0
13. time well estimated	1.8	0.9	2.4	1.0
14. organization of module	1.1	0.3	2.1	1.0
15. properly sequenced	1.4	0.6	2.2	1.1
16. quality of laboratory	1.3	0.6	2.4	1.1
17. time spent/value gained	1.3	0.5	2.4	1.1
18. faculty promoted learning	1.2	0.4	2.1	1.0
19. availability of self-instruction	1.4	0.5	2.2	1.0
20. cooperation of teaching team	1.0	0.7	2.3	1.0
21. exposure time (hrs)[3]	*	*	*	*
22. study time (hrs)[3]	*	*	*	*

N = 672

[1] Mean and Standard Deviation is based on the individual module score.

[2] Mean and Standard Deviation for all rated modules on that topic.

QUESTION AREA
SCALE

1 - 20

1.0 - 1.9 No improvement needed
2.0 - 2.9 Little improvment needed.
3.0 - 3.9 Improvement needed.
4.0 - 5.0 Considerable improvement needed.

[3] Due to data analysis problems this information was not meaningful. In future will be reported in whole hour estimates.

Appendix A

Faculty Ratings by Students
University of Florida/College of Dentistry

Faculty Evaluation For: Dr. Good

Date: 1 / 10/ 77

Number of Students Rating: 15

	TOPIC	Mean$_1$	SD$_1$	Mean$_2$	SD$_2$
1.	enthusiasm	1.5	0.6	1.9	0.9
2.	shows importance	1.4	0.6	2.0	0.9
3.	presentations	1.7	0.8	2.3	1.0
4.	objectives and goals	1.7	0.8	2.3	1.0
5.	aids retention	1.5	0.7	2.3	1.0
6.	stimulates learning	1.5	0.7	2.2	1.0
7.	clear requirements	1.4	0.6	2.3	1.0
8.	quality of tests	1.4	0.7	2.6	1.2
9.	patient welfare	1.4	0.8	1.9	1.0
10.	equal concern	1.4	0.6	2.1	1.1
11.	fair tests	1.4	0.6	2.3	1.1
12.	useful activities	1.5	0.8	2.4	1.1
13.	interested teacher	1.5	0.7	2.0	1.0
14.	promote progress	1.5	0.7	2.3	1.0
15.	seminars	1.3	0.5	2.3	1.0
16.	explains syllabus	1.4	0.5	2.1	1.1
17.	clinic availability	1.5	0.5	2.6	1.3
18.	receptive	1.4	0.5	2.6	1.3
19.	straightforward	1.3	0.5	2.1	1.0
20.	offers guidance	1.2	0.4	1.9	0.9
21.	seminar promptness	1.3	0.5	2.2	1.0
22.	accessible	1.6	0.7	2.4	1.0
23.	lecture quality	1.6	0.7	2.0	1.0
24.	clinically helpful	1.9	0.8	2.1	1.0
25.	academic advisement	1.2	0.4	2.2	1.0
26.	hourly contact[3]	4.5	*	3.0	*

N = 1374

[1] Mean and Standard Deviation is based on the individual faculty's scores.

[2] Mean and Standard Deviation for all rated faculty on that topic.

SCALE FOR TOPICS 1 - 25

1.0 - 1.9	No improvement needed.
2.0 - 2.9	Little improvement needed.
3.0 - 3.9	Improvement needed.
4.0 - 5.0	Considerable improvement needed.

[3] Key for Item 26

1 = 0-1 hours
2 = 1-5 hours
3 = 2-7 hours
4 = 7-9 hours
5 = 9 or more

RESIDENT PHYSICIAN PERFORMANCE ON A PATIENT MANAGEMENT COMPUTER SIMULATION

James V. Schultz, Ph.D., Robert S. Newsom, Ph.D., Steven Entine, M.S.,
Judith Neal, B.S. and Richard B. Friedman, M.D.

The Computer-Based Examination (CBX) Project at the University of Wisconsin-Madison in conjunction with the NBME and the ABIM has developed a patient management simulation model. This CBX simulation model is directed at summarizing a physician's performance as he/she obtains and interprets diagnostic data, reaches therapeutic conclusions, and modifies these conclusions on the basis of a patient's response. The physician's performance of the activities is, in part, a reflection of his/her patient management skills. Thus performance on the CBX simulation is assumed to be one measure of patient management skills.

The potential of such a simulation model for both physicians' training and assessment is considerable (1,2). Since the skills of patient management contain aspects of a problem solving activity the learning process for the acquisition of these skills include the possibility of error-producing actions. However, with the CBX model the physician user can pursue, through independent clinical actions, a variety of patient management skills without the constraints of real-life patient safety.

The physician user, seated at a standard computer terminal uses the CBX model to do a workup of a patient case. A simulated patient case begins with a presenting patient complaint (e.g., a 52-year-old white male enters your office complaining of right-sided chest pain). The physician user is then able to order any of over 400 tests, procedures, and consultations (e.g., history, physical, CBC, vital signs, SMA12, etc.), to treat the patient as an inpatient, outpatient or any combination, and can institute any of over 300 therapies or procedures (e.g., oxygen, penicillin, vincristine, digoxin, etc.). As the simulated (computer) clock moves ahead the progress of the patient is given and the test results are returned. Since a time axis is included, the patient's condition and with it the test results that have been ordered and reordered can change. These changes are dependent on the therapies that may or may not have been started in conjunction with the normal course of the patient's underlying condition.

A preliminary study, directed at determining the workability of the simulator models under a large scale testing situation was conducted in September - October, 1976. Specifically, the study was concerned with obtaining information on two focal questions:

1. Does the simulation model function in a realistic and usable manner?
2. Does resident physician performance on the simulator in any way reflect real-life performance with actual patients?

Prior to the initial interaction with the simulator the physician user is instructed in the use of the terminal, the options available in the model and the 'psychological stage' of primary care agent is set via a short, written orientation.

Analysis of physician performance is based on whether they handle the various patients in a 'correct' manner; that is, a manner compatible with what would be done in real life. Consequently, the users are instructed that they should seek to minimize the following factors: a) any risk to a patient in terms of pain, complication, and/or mortality risk potentialities of the tests ordered or the therapies instituted, b) the cost to a patient both of the workup (tests, consultations, therapies, etc.) and hospital charges, c) the length of stay for a patient, d) the time to start of corrective therapy(ies), and e) the time spent on a particular patient case. However, the physicians are instructed to be aware that by excessively cutting corners to minimize the above factors they will subject the patient to other risks which may be equally as bad, (e.g., either a missed or only partial diagnosis as well as inappropriate therapeutic interventions).

For this study, approximately 81 first, second and third-year resident physicians at the University of Wisconsin-Madison and Mt. Sinai-Milwaukee hospitals used the CBX simulation model. Each resident spent a four-hour session at the computer terminal doing a workup of at most six patient cases. These patient cases covered the disorders of myocardial infarction, diabetes, and pneumonia. In addition a case with a normal patient was available to allow the residents to practice using the simulation model. Each resident completed a pre- and post-simulation encounter questionnaire. The Likert scaled questionnaire covered the residents' attitudes toward computer simulation and their ability to appropriately use the CBX model. Finally, a protocol for each patient case completed by a resident was retained in the computer, indicating the sequence and (simulated) time of all tests and therapies ordered.

Each of the cases, of which ten were used in this study, was developed by physicians at the University of Wisconsin-Madison using actual charts of the cases for a model; each script is then checked for continuity and consistency via a flow-charting procedure by the CBX staff. The flow-charted script then proceeds to expert committees of the NBME and ABIM for final approval and correction before its being programmed into a PDP-15 computer. A computer printout of the case is finally compared again to the flow charts for accuracy before it is opened for initial user testing. The nature of such interactive programs results in each user developing a idiosyncratic pathway through the case; consequently, the cases are not considered finalyzed until such time as non-predicted test or therapy interactions do not occur. In this regard the CBX simulation transcends real life where ill-advised tests or therapies would surely result in high risks of pain, complication or mortality and the attendant consequences these might bring. The freedom to attempt what in real life might result in a tragic test or therapeutic intervention on the simulator is obvious to the user, yet of the hundreds of users who have interacted with the model all of these have maintained courses of action which indicated a strong unwillingness to treat the cases as anything other than a real-life situation. While, of course, this simulation might be viewed as an elaborate game, no user to date appears to have reacted to it as such. It is assumed that this reaction is a result in part of a situational motivation generated by the user's knowledge that his/her patient management skills are being examined. Perhaps if the model was available only as a learning device different approaches and behavior on the patient cases would be observed.

Results

Data collected from the resident physicians generally indicated a positive affirmation for both of the study's focal questions. Table One contains the performance data on the number of completed patient cases in the four hour terminal session. As can be seen 31% completed 5 or more cases while 46% completed 4 or more. Since the vast majority of the residents had little or no previous experience using computer terminals let alone experience using a patient management simulation model the fact that almost half of them completed 4 or more cases is encouraging.

The average time, standard deviation and number of residents completing each script is given in Table Two. Importantly the three patient cases with the largest mean times (68, 60 and 51 minutes for cases 36, 38 and 30, respectively) were for a majority of the residents, the first case completed. Consequently, it appears that much of the time spent on this first patient case was devoted to learning and becoming familiar with the mechanics of the simulator model.

Analysis of the attitude data indicated that the residents were generally unable to decide how they felt about the simulator model. The average score on the five point Likert scale with 1 = disagree, 3 = unable to decide, 5 = agree was 2.73 (SD = .49). Also, the pre- and post-scores on the scale were pre- average = 2.83 (SD = .54) and post- average = 2.85 (SD = .55), again indicating an 'unable to decide' orientation. Though not conclusive, these results are encouraging when one notes that the residents who used the simulator were told they were being tested and it is highly doubtful that for any test a high positive attitude would be obtained. Further, this was a first time experience at using such a complex simulation model and as such, one would probably expect at best a wait and see attitude. A rank ordering of twelve attitude questions that asked the residents to react to specific aspects associated with their terminal sessions revealed the following: the highest responses were for a) able to order tests using the simulator, b) felt involved with patient cases, c) could grow to enjoy and learn from the simulator, and d) the problems approximated real life. The lowest responses were for a) the simulator did not hinder my ability to think clearly, b) could react to patient change using the simulator, c) personal learning took place in the encounter, and d) the simulator indicated my strengths and weaknesses.

Analysis of the residents' protocols revealed that they ordered the appropriate tests, reached correct therapeutic conclusions and generally treated the simulated patients much as a physician would treat a similar real-life case. Table Three presents a comparison of some of the tests ordered and the therapies started for each of the disorders used in this study. The simulation column shows the percentage of residents who ordered the given test or started the given therapy while using the simulation model , while the percent in practice column shows these same figures for a group of actual patients at the University of Wisconsin-Madison hospital. Clearly, the table indicates that the residents' performance in simulation generally reflects real-life performance on actual patients. Further ongoing research will provide more specific evidence as to the efficacy of this simulation as a measurement and training device.

References

(1) Friedman, R.B., "A Computer Program for Simulating the Patient Physician Encounter," J. Med. Educ. 48(1): 92-7, January, 1973.

(2) Friedman, R.B., "Clinical Simulation in Medical Education - The Wisconsin Experience," Computers in Biomedical Education (E. Deland, Ed.) (in press).

This work was conducted by the Computer-Based Examination(CBX) Project at the University of Wisconsin-Madison and was supported in part by the National Board of Medical Examiners and the American Board of Internal Medicine. The authors are all on the staff of the CBX Project-University of Wisconsin-Madison. Dr. Richard B. Friedman is Director of CBX and an assistant professor in the Department of Medicine, Dr. James Schultz is Assistant Director ofCBX and an associate scientist in the Department of Medicine, Dr. Robert Newsom is a Project Associate, Mr. Steven Entine, Mr. Sunming Cheung, and Ms. Judith Neal are Project Specialists in the Department of Medicine.

TABLE ONE

Distribution for completion of patient cases in 4 hr. session.

Scripts Completed	N	%N	Cumulative %
1	11	.14	1 or more - 1.00
2	19	.23	2 " - .86
3	14	.17	3 " - .63
4	7	.09	4 " - .46
5	9	.11	5 " - .37
6	21	.26	
	81	1.00	

TABLE TWO

Average time, standard deviation and number of residents completing a patient case.

Patient Case	Average Time	Standard Deviation	N
36 (MI)	68.0	47.59	43
38 (P)	60.3	26.30	44
30 (MI)	51.9	32.54	33
72 (P)	40.7	55.57	25
49 (D)	36.7	26.46	34
40 (MI)	36.6	27.40	35
42 (P)	35.4	23.93	31
70 (D)	26.7	14.41	26
46 (D)	26.2	15.87	35

TABLE THREE

Selected summaries of tests and therapies for the disease states ordered by residents in simulation and practicing physician(s) working in a university hospital.

	Pneumonia		Diabetes (ketoacidosis)			Myocardial Infarction		
Tests (Day 1)	% in practice	% in simulation		% in practice	% in simulation		% in practice	% in simulation
HX	+	.80	HX	+	.87	HX	+	.84
PE	+	.95	PE	+	.98	PE	+	.98
VSG	1.0	.50	VSG	1.0	.59	CXR	.78	.85
Sput. C & S	.89	.61	SMA12	.40	.43	EKG	1.0	.99
Blood C & S	.56	.62	CXR	.90	.70	VSG	1.0	.63
U/A	.89	.36	ABG	.60	.89	lytes	.78	.72
CBC	.56	.83	Glucose	.70	.82	LDH	.89	.60
ECG	.89	.65	Urine C & A	.90	.61	SGot	.89	.63
ABG	.33	.84	lytes	.50	.81	CBC	.55	.67
CXR	.77	.93	CBC with Diff.	.70	.60	CPK	1.0	.82
Therapies (Day 2)								
O_2	.44	.68	Insulin	1.0	.68	O_2	.67	.75
Fluids	.44	.35	IV-Fluids KCL	.90	.75	Lido-caine	.56	.69
Some type antibiotic	.50	.46						

+ History and Physical were not included in written log of University Physician(s) activities.

CAI SYSTEM TRANSPARENCY -- ONE SOLUTION

James A. Schuyler, PhD
University of the Pacific, School of Medical Sciences
(PO Box 7999, San Francisco, CA 94120)

CAI courseware transport has many levels -- personally I am working on problems which may be regarded as "technical". There are also many organizational problems, including review, validation, cataloguing, distribution, and updating of courseware. And, although it's hard to pin them down, there are personal problems involved (of the "who wants to use somebody else's lesson" variety).

From my viewpoint, based on experience and current research here at UOP, the "language-barrier" problem (on the technical level) is grossly overrated. I am tempted to say there is no language problem!

Language barriers

The problem of language is a technical problem. We attack such problems because it's easy to "get a handle on them", and this one is no exception. It really reduces to a question of translation. Translation is accomplished at two levels: 1) the level of source statements, and 2) the level of educational intent.

Source statement translation is no problem. Numerous individuals have found mappings and created translator programs to go from one language to another. The more general approach being taken here at UOP is to build mappings (rule-sets) from source CAI languages into a "core" CAI Design Language (and inversely from the core into other languages). The "core" language is all-inclusive, containing functions needed to translate any CAI courseware. Though we did not design it as such, it could be viewed as a "super" CAI language, on roughly the same level as TUTOR-IV. Once mappings are created to and from the "core" and a CAI language, the source-statement translation problem is solved.

At the level of educational intent, there is more difficulty in translation. Let me digress, then return to this point later.

The Transparent System

The ultimate goal of work here at UOP is to totally eliminate the technical barriers of language. Authors will be able to work in whatever CAI language they feel comfortable with on our computer -- their courseware will be rule-translated to the "core" language for testing on a local system, and rule-translated to a "target" language for delivery to remote (student) sites. In fact, there's not even a compelling reason to strive for standardized languages -- as long as the implementer of a new language can provide a translation rule-set, her or his work can be translated to the "core" and from there to any target. The day of "personal" languages may not be far off. On the CAI Design

System we expect an author who knows TUTOR and who has a graphics terminal (one equivalent to a PLATO terminal, at least) available, to work in TUTOR, with the translator functioning out-of-sight.

Translating Educational Intent

If you examine the example of the TUTOR author presented above, you will come to one stumbling block. It is possible for courseware to be source-transportable, but not be executable on the target system for lack of a proper presentational medium (terminal). Even on PLATO itself, one frequently encounters a lesson which requires touch-panel, which your terminal may not have; it's very disappointing. It's apparent that we need to structure our courseware world in levels. Let me suggest that we begin by listing some of the parameters of the media (terminals), and using these to characterize the levels of interaction possible:

Level:	Examples:	Comments:
Audio systems	Audio output or input	
Positional inputs	Touch-panel or light-pen	Characteristics include grid size, coordinates.
Font or dot graphics	PLATO, TICCIT	Character sets may be defined, color set, blinking, etc.
Vector-graphics	CRT or plasma displays	Dot-by-dot grids for the most part.
Character grids	simple CRT's	Individual character positions on screen may be "randomly" addressed.
Serial, print oriented	teletype replacements	Print only, both in and output.

[See also Avner's comments in following paper.] It is my view that every courseware catalogue published should distinguish among these various levels, as they are often the most important technical barrier to transport.

There is, of course, the possibility of "hand" translation from a higher to a lower level (see table). I think that 25% to 50% of the TUTOR lessons I've ever looked at utilize "cutesy" graphics which could be easily eliminated or moved to paper. The decision to cut such frills, however, must be made by a translator familiar with the educational intent of the courseware and its actual use with students.

So, back to the problem of educational intent. In the long run we CAI authors and system designers should get away from "programming" and move toward representations of the interactive process we wish our students to experience. I think this will mean the writing of courseware in a "standard" format (flowcharts?, an interactive "language"?) which is then interpreted by a CAI system using instructional strategies most appropriate for individual students. [See Camstra's paper, which follows.] And it's my view that now that we can stop fooling around with source-language programming we'll be able to get down to the real work of transporting educational resources.

Courseware Transport--When and How?

R. A. Avner
Computer-based Education Research Laboratory
University of Illinois at Urbana-Champaign

Commercial vendors and owners of CAI systems have come to a realization of the difficulties of producing effective courseware with limited resources. It seems only reasonable that duplication of effort be avoided by a sharing of this expensive but necessary component of computer assisted instruction. Anyone in such a position of need looks longingly at the riches suggested by the latest edition of one of the indexes of CAI materials.

The seeming advantages of being able to freely access materials from every CAI system are so compelling that it is easy to overlook some cogent reasons for not providing universal transport mechanisms at this time. The position of the devil's advocate in this matter is roughly outlined here.

Basic to the issues of standardization and transportability is the matter of equivalence of function. Whether one is talking of languages to be used in international commerce or circuitry for color television, standardization is reasonable only when the alternatives serve essentially the same function. Despite its all-encompassing title, all CAI is not equivalent. Most of us who have spent substantial amounts of time converting materials from a teleprinter-oriented system to a graphics-oriented system (or vice-versa) have been struck by the fact that these are really different media and that materials that make effective use of the features of either orientation rarely transfer well to the other. Even the differences between random- and serial-access CRTs can wreak havoc. What at first view appears to be a case of one-way compatability (where courseware developed on a "less-sophisticated" system can be used on a more advanced system--but not the reverse) often requires substantial and expensive revision just to insure that the transferred material functions.

A second requirement for courseware transport is the availability of something to transport. The current universe of CAI materials is not accurately identified. Summaries of "available" materials are dependent on the good intentions of those who contribute information. It is as if Books in Print included every idea and rough draft ever thought of by every prospective author. All too frequently materials listed as "available" are either not available, incomplete, or of unacceptable quality. There are indeed good materials that deserve wider audiences but they are not available in the quantities suggested by available documentation.

Having said this, I must also state that I would be among the last to support a return to Balkanization of the field. Courseware sharing is a short-term objective that has potentially favorable long-term outcomes.

First, systems that are in fact equivalent media do exist. Sharing conventions should be established wherever the availability of high quality materials makes exchanges mutually advantagious for users of such systems. This sharing provides an empirical basis for identifying language features that warrant adoption as well as cutting costs of courseware development. Where media are equivalent and languages are similar, sharing can be done easily. Unfortunately, many workers tend to minimize media differences and assume system equivalence solely on the basis of CAI language. The fact that two systems both use BASIC, for example, does not automatically mean that they are the same instructional medium. If the same language is used for different media there is actually the chance that access to one medium is degraded by use of a language inappropriate to its structure.

Efforts to facilitate courseware exchange also force attention to another area of communications that has been long deficient in CAI. Documentation of courseware is generally so sketchy that there is seldom any way short of viewing material to tell what pedagogical techniques are actually used. Although coding is a highly visible cost, studies at the University of Illinois on distribution of effort by experienced CAI authors indicate that the major part of total production effort is in specification of logical and pedagogical approaches rather than in generating code specific to a given CAI language. A substantial part of the value of courseware sharing thus accrues simply by being able to make use of the instructional kernel of imported materials. Needless to say, adequate documentation would save much effort just by adding to a designer's repertoire of approaches of proven effectiveness. Reports and studies that contrast "standard" instruction against "CAI" without indicating precisely what the student experiences are worthless either as a means of communicating effective design approaches or as a basis for deciding if material is transportable.

Courseware sharing is a partial solution to a pressing problem. Materials written for equivalent media can and should be shared. However, CAI is not a single medium. Efforts to facilitate transportability of courseware can have positive effects by focusing informed attention on software features and pedagogical technique that deserve wider use. This transport of tools as well as products is, in the long run, more likely to contribute to the viability of CAI than short-term savings in courseware production.

Problems in the Future of Courseware Sharing

Stephen K. Lower
Simon Fraser University
Burnaby BC Canada V5A 1S6

Authors and royalties.

The authoring of CAI material will bear many similarities to the authoring of textbooks or films; almost anyone will potentially be able to do it, and many will dabble at it-- but the bulk of the best, comprehensive material will be created by a small number of people who are skilled communicators and teachers-- and (in third place) programmers. There will, in addition, be more emphasis on group and cooperative projects.

Royalties will not generally be the primary motivation for CAI authors, any more than they are for most textbook authors. For one thing, the large amount of "free" material available, although it may be largely third-rate, will tend to depress the market value of all CAI material. Nevertheless, people will be willing to pay for excellent, well-documented and comprehensive CAI packages that promise to do a specific job. Payment will probably be through a licencing arrangement.

Publishers will probably not produce CAI materials themselves, since the extremely high costs of employing people specifically to write CAI would be difficult to recoup. More likely, they would act as agents for authors; they could promote and distribute the material, and would be better able to vigorously protect the owner's rights.

My own motivation as an author is a desire to communicate-- and the challenge of trying to do so more effectively than anyone else. Financial reward would be frosting on the cake. Nevertheless, I am a bit taken aback by the letters I sometimes get, asking for me to "send us all of your CAI programs", or "send us everthing that is free". Granted that most of the senders of such letters usually have only the most vague idea of what CAI is about, and only the most primitive equipment for delivering it (one wanted me to send my complete works on paper tape!), I wonder how such people would respond to a similar request to "send me a photocopy of the manuscript of your new textbook"? Certainly a project that has consumed years of an author's time and experience is worthy of at least token payment.

Language standards.

Language *standards* are far more important than having a *standard language* (see section on translation, below). What is important are *functional* standards: what kinds of things a language should be able to express, and at what levels of coding. Whatever standards do evolve will probably be *de facto* ones, dictated mainly by the desire of others to utilize the best courseware available, which will of necessity be written in a high-level, powerful language. "Simple" languages will always be around, but few educators will find as much appeal in 100 1-hour courses written by 100 different authors (using PILOT in store-front computer centers?) as they would in one 100-hour course by a single author-- and I can't imagine one person writing this much material (and remaining sane) using a primitive authoring language.

Manufacturers will probably exert little positive influence, since they are unlikely to see much profit in developing languages or in improving them. Witness Coursewriter III, which has been moribund for years, or IDF, which, being tied to a specific kind of system, is not likely to evolve much. The driving forces behind language development and improvement will come from authors and users. Similarly, "national" authoring languages, in the absence of networking or some scheme of forcing authors to use them, are unlikely to have any major effect other than the promotion of language standards.

Networks.

I don't see much future for networks for CAI, at least in the ordinary sense. Although the costs of computing and data storage continue to decline, the costs of data transmission will remain steady or rise. In CAI, "line time", not number of bits transmitted, is the crucial cost factor. In addition, there is a trend to smaller systems-- clusters of terminals, or stand-alone "intelligent" terminal systems. These systems may dial in to a network to get daily updates on programs, but will not depend on a network for regular CAI operation.

Networking may be practical in a flat, densely-populated state like Illinois, but in British Columbia, with far lower population density and the most rugged terrain in North America, where telephone communication to most regions is by expensive microwave relay, networking would be out of the question.

Language translation.

Not only feasible, but it is being done: PLANIT is being translated into a minicomputer-based "CAI Assembly Language" in Delft, Holland. Coursewriter is being translated into PILOT by WWSC, into IPS at Simon Fraser, and into IDF by H-P. As long as two languages have the same general functional capabilities, they can be translated; modern text editors and string-processing languages like SNOBOL make this a relatively simple matter, requiring a minimum of recoding.

I believe the possibilities of translation are rather underrated; given enough good course material written in any language, there will be incentive to translate.

For us at SFU, it means that we can use our own language that meets our own requirements of power and sophistication, and still not isolate ourselves from users of other languages such as Coursewriter (and hopefully, TUTOR).

Validation and review

My experience has been that the "effectiveness" of any instructional medium depends as much on the way it is used-- on the sensitivity with which it is interfaced to the course as a whole-- as it does on the inherent "quality" of the material itself. I am therefore rather dubious of many "validation" studies I have seen that do not take these factors into account. At the present time, *review* may be a more practical approach, although it will be a difficult task, far less casual than the review of a textbook. One of the real problems is that the newest and most creative ideas may be demonstrable only on the most sophisticated systems, and thus inaccessible to most potential reviewers. Of course, one can go to the opposite extreme, and insist, as CONDUIT does, that authors stick to the technology of ten years ago (FORTRAN and BASIC), thus sacrificing innovation to the god of transportability. At the present time, it would seem far more important to simply find out who is using CAI in a domonstrably successful way, and how it is being used.

Sharing courseware

We college professors are all basically *prima donnas*, unwilling to expose our students to any but the best (our own!) materials. Pressed by circumstances, we will (grudgingly and "temporarily") use someone else's text-- and we might even use another's CAI program, but only if we felt that we had "control" over it, or at least thorough understanding of it. Other teachers may feel roughly the same, but are generally under far more pressing "circumstances"-- for instance, they are actually expected to teach. Courseware sharing is not likely to be widespread until authors pay far more attention to documentation, so that the teacher-user can acquire an accurate feeling for what the courseware contains, how it works, and how it can be modified to meet special needs.

Significant problems

We have always regarded the lack of a really powerful, generally usable *authoring language* as a major impediment to the design and documentation of CAI. Having given up on Coursewriter and APL, we finally designed our own, incorporating what we feel is the best of these two languages, with many of the features of TUTOR and other proven languages-- together with numerous ideas arising from our own practical experience in using CAI. We offer it not as an "ideal" language, but as an example of what can be done using present design practice and technology.

An *insignificant* problem-- in the long run, is small-computer capability and terminal hardware. The recent announcement of a 92K monolithic memory chip and the promise of better to come should make us raise our sights about CAI, and concentrate on the *real* problem that technology will never solve: getting the best teachers to write outstanding courseware.

TRANSPORTABILITY AND SHARING -- THE NEED FOR STANDARDS

J.W. Brahan
Division of Electrical Engineering
National Research Council of Canada

While a variety of factors can be cited as inhibiting the sharing of computer-based courseware, the major technological barriers are closely related to a lack of standards for both hardware and software, with the latter being the main area of concern. Problems of hardware interconnection have been greatly reduced through the efforts of bodies such as the Electronics Industries Association (EIA) and the International Organization for Standardization (ISO) with its associated national bodies. Standards remain to be developed or updated however, in a number of areas of information interchange which are of concern in Computer-Aided Learning (CAL) hardware design. Of particular importance is the definition of standards for control of the complex graphic and multi-media terminals made possible as a result of the micro-processor. Work currently underway within the standards organizations suggests that we can anticipate a resolution of these problems for the general case which will satisfy most of the specialized requirements for Computer-Aided Learning. Hence it is reasonable to conclude that current hardware problems specific to CAL applications can be considered primarily questions of functional and performance differences between terminals. It must be noted however, that this is valid only if CAL hardware conforms to the published standards and if the appropriate standards organizations are kept informed of any special CAL requirements.

Insofar as software is concerned, the primary deterrent to courseware transportability is the lack of a generally accepted programming language for the development of educational materials. To promote transportability, many projects have adopted the policy of using a general-purpose language such as BASIC or FORTRAN for which published or de facto standards exist. In addition, programming is often confined to a basic subset of the language. For most CAL applications this is far too restrictive and while a high degree of transportability can be achieved, it is usually at the expense of quality and performance of the courseware program. Performance often is also limited by the lack of facilities to support the courseware author in preparation of materials, and to support the delivery of materials on the complex terminals which can be required in many CAL applications. At best, this approach to achieving transportability should be considered an interim measure.

The integrated-system approach as exemplified by systems such as PLATO, TICCIT, the IBM-1500 and others provides much more in the way of specialized software to support preparation and delivery of courseware. Each user has the same terminal hardware and uses the same programming language. Thus, within the confines of the system, transportability ceases to be a problem and any courseware developed for the system can be made accessible to any user who has access to the system, either through a network or a local replica. The integrated system, in effect, defines a set of "local standards" to which each user must conform. The constraints imposed by these standards have the advantage, within the system community, of removing any technological impediments to courseware sharing, of facilitating the interchange of information between courseware developers and of simplifying software development and training problems. There are, however, several disadvantages. The user must commit himself to the "standard" system terminal whether or not he needs all of its features. He cannot use a different terminal which may be available as a result of other requirements. Updating of system hardware to take advantage of new technological developments can be expensive. The user does not have easy access to materials generated outside of the system nor can he readily transfer his materials to other systems.

If computer-based courseware is to exploit effectively the potential of the computer, be disseminated beyond the confines of specialized systems and be economically viable, software standards must be developed and propagated. Such standards must take into account a wide range of user needs and be applicable to the variety of systems and equipment which can reasonably be expected to be used for CAL. It was with this requirement in mind that the National Research Council's Associate Committee on Instructional Technology initiated the work towards development of a common programming language for CAL. In 1972, the Committee published a functional specification for a CAL programming language which presented a consensus of opinion of a group of active CAL users and was used as the basis of a contract for the development of a detailed specification. The contract was completed in 1974 with the publication of an Author Guide and Specification Manual. This was followed by a preliminary implementation of NATAL-74 which is now operational on a DECsystem-10 in the NRC laboratories. While a few features of the NATAL-74 Specification remain to be added to the implementation, initial testing has begun. To date, this testing has consisted primarily of translating several CAL programs from other languages and has provided only a limited assessment of the language. However, it has given some indication of the capabilities of NATAL-74 for easing the task of the course author in exploiting the capabilities of the computer. The testing has also demonstrated that CAL programs can be adapted to different terminals (within practical limits, of course) by the system without modification by the course author. Assessment of the effectiveness of NATAL-74 as a vehicle for transporting programs from one system to another will have to await its implementation on other machines. However, the potential benefits and the interest which has been shown by CAL users indicate that this is a task worth completing.

Translation is often proposed as a means of transporting CAL courseware programs from one system to another. The translations which have been done as part of the NATAL-74 project have demonstrated that most of the effort goes into understanding the source program. Documentation of courseware is rarely prepared with a view to meeting the requirements of translation which, in many cases, becomes a difficult and costly task, particularly if it is not possible to run the courseware in its original version. For translation to be a viable means of transportation, ideally, courseware would be defined in a system-independent language and that definition would be the source from which the original and translated programs would be written. Unfortunately, the development of a suitable language for this purpose and getting people to use it would seem to present even greater problems than development of a common programming language.

Effective sharing of courseware, whether or not translation is involved, requires that the courseware be accompanied by good documentation. Thus a necessary component of software standards for CAL consists of documentation guidelines which define the information required to provide for understanding, maintaining and using computer-based courseware.

While sharing of courseware is affected by more than the technological issues, these must be resolved before we can have widespread dissemination of computer-based courseware. The principal aspects of the problems to be resolved include development and specification of CAL hardware in compliance with accepted general standards, development of a standard programming language and development of effective courseware documentation standards. Sharing will not come automatically once these issues are resolved, but they are prerequisites to resolving many of the other problems.

"The Problem of Natural Language Barriers"

Bert Camstra, Center for Research into Higher Education (COWO)
University of Amsterdam, Netherlands

There is a problem in the transport and sharing of computer-assisted instructional material that people in English speaking countries are seldom aware of. That is the problem which arises in international sharing: the diversity of languages. By this we [computer] freaks have come to mean computer alnguages, but that is a problem people are very much aware of, and is something I will not discuss. I will concentrate on "mother languages", various natural languages.

As in the flow of scientific information (cf. Bung, 1976) and of instructional materials like textbooks, the flow of CAI materials depends to some extent on (an in most cases irrelevant) an aspect -- namely, the language it is written in. If we assess the current situation in CAI we will acknowledge the facts that:

1) English is the major world language,
2) A large majority of CAI materials is produced in English-speaking countries,
3) Most English speaking CAI customers cannot be expected to be fluent in a foreign language,
4) There is an enormous variety in the character of instruction in various countries.

These facts have a number of consequences for the sharing of CAI materials between English speaking (ES) countries and non-English-speaking (NES) countries.

Facts 1 and 2 result in a potentially large flow of CAI materials from ES to NES countries. Fact 4, however, puts a restraint on this flow, because in many cases the CAI material requires local adaptation, not only because of hardware, software and translation from English to the local language, but also because of the character, atmosphere and goals of the local instructional traditions and systems.

Facts 2 and 3 limit the flow of CAI materials from NES countries to ES countries (e.g. "who wants my course in experimenting for psychology sophomores if he cannot read Dutch?"). Fact 1 implies that students in many NES countries do read English to some extent. As we all know however, CAI is RESPONSIVE instruction almost by definition. This means that computer assisted students must not only be able to read English but also to PRODUCE it. Only in a very few countries can the majority of university students produce understandable English sentences (e.g. Holland, Germany, Denmark, Sweden, etc.). Even then, the fact that English is a foreign language results in two problems that current CAI programs cannot generally cope with: the fact that non-fluency is expressed in gross syntactical errors, and the fact that it is expressed as limited vocabulary. While the first of these is sometimes not problemmatical (because of the "stupidity" of the answer checking routine), the second problem is solved by the speaker's circumlocution rather than direct naming -- an insurmountable problem for the currently "stupid" CAI programs.

The conclusion of my assessment so far must be that international sharing of CAI materials cannot be a major affair in the next years. Apart from any solution that perhaps will be found for hardware and software transportability problems, there will still be a need for translation if different languages are involved in the sharing.

Now on to potential solutions: First, I will NOT propose to found an international CAI translation center somewhere in the world. Even if it would be possible to get it started, it would surely run into massive problems and would not be successful. Even thinking about explaining why makes me exhausted.

Rather, I see two potential solutions, both with mich wider scope than the present problem. The first is to "Make CAI Smarter". The essence of this development is to make CAI generative. I have described this elsewhere (Camstra, 1977). While relatively unsophisticated "smart" CAI programs can manipulate preprogrammed texts that require translation, the more advanced "smart" CAI programs GENERATE verbal output. It has been shown that it is possible to transform an English-language generator into a Dutch language generator and vice versa (Kempen, in preparation). Moreover, I am sure that it is less work than translating all utterances that such a generator might produce. Obviously, creating intelligent CAI is a mammoth task for the next ten years and does not address the current transportability problems. Nevertheless, it is an approach to circumventing the problem of different natural languages worth pursuing.

The second potential solution (independent of the former) is probably not new, and moreover may be a solution for other problems, such as adapting CAI materials to different hardware, or software or local instructional emphases or character. That is to describe a program or course at a level that is THE LOWEST LEVEL INDEPENDENT OF A PARTICULAR IMPLEMENTATION. In many cases this description would take the form of a detailed FLOW DIAGRAM, annotated by the author with statements, warnings, advice and other documentation. In this conception, it is not courseware which would be transported but the ideational framework on which it was based. Clearly, the CAI material would have to be reprogrammed each time, and one of the objects of the transportability movement is to find ways to make that unnecessary. Nevertheless, describing courseware at the lowest implementation-independent level might be a way to reduce the programming time drastically. Moreover, a local product is apt to be more successful in a strange computer environment more suited to the particular student population and more appreciated by the local teacher, while at the same time overstepping the foreign language barrier.

In summary, I hope my two ideas will be discussed at this conference:
1) Make CAI smarter,
2) Transport descriptions instead of programs.

Bung, Klaus. "Educational Technology in Germany and its European Context", in: A. Howe & A.J. Romiszowski (eds.) INTERNATIONAL YEARBOOK OF EDUCATIONAL AND INSTRUCTIONAL TECHNOLOGY, 1976/77, London: Kogan Page, 1976.

Camstra, Bert. "Maek Computer-Assisted Instruction Smarter", to be published in COMPUTERS AND EDUCATION, 1977.

Kempen, Gerard. in preparation.

A Case Study Using MUMPS Examples [Condensation]

Richard F. Walters, PhD
School of Medicine
University of California
Davis, California

In 1973, Boehm[1] stated that about seventy-five percent of the cost of computing can be attributed to software, and that this percentage will increase to a figure closer to ninety percent over the next decade. In education, the exchange of teaching materials becomes particularly important, since software involves not only programmer expertise, but the contribution of faculty and researchers in educational strategies.

Exchange is enhanced by commonality of language, and particularly when that language is implemented on the same type of computer. However, even in the most optimal conditions, exchange is not easily accomplished.

Political Factors

If a transaction is not to the benefit of both parties, it is doomed to failure. It is self-evident that the recipient stands to benefit, but the developer must also gain from the exchange, either financially, politically, socially, personally (satisfaction of ego) or legally, in fulfillment of contractual or fund support obligations.

The Lister Hill Center for Biomedical Communications supported exchange by making available a few key teaching programs to medical students throughout the country, selecting three hosts, one of which was Massachusetts General Hospital (MGH), and linking these to a national timesharing network. This experimental design succeeded beyond the designers' wildest expectations because it was of benefit both to provider and user, with exposure and quality software as incentives.

Technical factors

Naturally, the language in which a program is written is a major element required for sound technical evaluation. Of equal importance is the manner in which the language is implemented on development system vs. the system at the receiving institution, size of the program, and its dependence on features not defined in the target language, and dependence on special system subroutines or other supports. There must be expertise available to solve problems at either end.

MUMPS was originally developed at MGH, and because of attractive features was imitated elsewhere. Since it was interpretive, the question of self-modifying code made program transfer especially complex. As described by O'Neill[2], the Department of Health, Education and Welfare's Health Resources Administration and the National Bureau of Standards elected to support the development of a standard MUMPS. The program emphasized the

180

active cooperation and commitment of the MUMPS community. Supporting agencies refrained from taking a leadership role. This design has proven effective in terms of success achieved in new implementations. In addition, the transparency of the national timesharing service used proved to be a great benefit in the medical environment. Remaining problems can now be addressed by the MUMPS User's Group, facilitated by the creation of a semi-automated translator.

Documentation

There is a profound difference between software for internal use and development with the prior supposition that it will be distributed. Sometimes, but not always, a local program can be documented sufficiently to enable transfer. The MUMPS Development Committee's efforts to define documentation standards and demonstrate models for such standards are exemplary.

The Process of Exchange

These have all been *a priori* conditions for software exchange. If all are satisfactorily resolved, then exchange may begin.

Agreements for Exchange

Most important factors are those of economics -- there is no "free" exchange. Costs of preparation, duplication, dissemination, documentation and validation must be borne by one or both parties, and an agreement is critical. A second consideration is the legal effect of the use of the program. In medical applications this may be more critical, but failure of a program may result in serious economic (or other) consequences. Liability for such failure should be established at time of exchange. The degree of unspecified support is also a question; to what extent will developers be required to respond to telephone calls, letters or requests for information and assistance? In addition, parties should agree upon ways in which program deficiencies will be handled, whether real or the result of misunderstandings.

Lister Hill's contribution has been through direct support of a pilot study involving transfer from MGH to the University of California at Davis, and the MUMPS User's Groups has addressed this area through its subcommittee on utilization.

Technology of Physical Transfer

The number of ways in which a set of computer instructions may be "read" by a machine may provoke difficulty in obtaining commonality. A promising solution is the use of national timeshare systems to link minicomputers so source code may be transferred at attractive,economic rates. Transfer between MGH and UC Davis is done through the national timeshared network also used by students. The MUMPS User's Group has also identified several local institutions to serve as distribution centers.

Documentation Support for Exchange

The receiver needs to know the architecture of the software, requirements for auxiliary data and files, reliance on overlay, chaining, virtual memory or other elements. A series of validation routines is available for testing new implementation of MUMPS.

Post-Delivery Considerations

The process of program exchange continues. Is the recipient institution authorized to modify the code that has been received? To what degree? The inevitability of disputed merits of user-provided updates will also lead to differences that are better anticipated and resolved before they actually arise. The manner in which user-feedback is retrieved by the developer should also be spelled out. Even the support of documentation during program evoluation is rarely followed (through distribution of page inserts for manuals).

MGH is also working on the questions of remote authorship and has some pilot studies under way. Lister Hill is actively considering strategies for continuing feedback of student evaluation. The MUMPS User's Group is establishing mechanisms for revision of documentation and teh Development Committee is addressing the area of language evolution and its support.

Conclusions

The need for user's groups is important. The need for networks to provide continuing media for communication, test, transfer and update is one of the single most important factors, throughout the entire development process. The support of special interest groups to define areas in which new programs are required is important. Finally, the efforts required to develop language standards and to devise means whereby adherance to these can be established looms as one of the most important elements in program transfer. The remarkable success of the MUMPS community in this regard cannot be ignored and as the language approaches the American National Standards Institute's (ANSI) approval through the relatively untested Canvass Method approach, its progress should be watched closely by those wishing to adopt similar standards for special or general purpose languages that will inevitably evolve in the future.

References:

1. Boehm, B. "Software and Its Impace: A Quantitative Assessment", DATAMATION, Volume 19, No. 5, pp. 48-59

2. O'Neill, J. T. "MUMPS Standardization", MEDICAL INFORMATICS, (Companion paper in this issue).

POSSIBLE FUTURES FOR COURSEWARE SHARING - A ROLE FOR INSTITUTIONS

Robert G. Votaw
University of Connecticut

I. Introduction

The frame of reference of this paper is overall development of computer-based education in the health professions and the roles that may be played by institutions. Of this, courseware sharing is an important subset. My purpose is to cause reflection on fundamental assumptions about priorities which express in our individual and joint decisions regarding the allocation of time and resources related to CAI. If a problem is conceptualized incorrectly or on an inadequate scale, the return on investment of much time and effort, yours and mine, is less likely to be satisfying.

As my fundamental assumption, I will hypothesize that over the next quarter century health professions education will relax into a new "steady state" educational model in which computer-based education (CBE) will assume a very large role in the health educator's armamentarium. Parenthetically, I also assume CBE will not totally replace other traditional and non-traditional means of instruction. In order for this hypothetical situation to be realized, three fundamental conditions must obtain. First, there must be individuals who need computer-based education materials and who, we can assume, will be unable to produce locally all materials necessary to meet their needs. These individuals must be willing to share courseware produced by others. Second, there must be courseware available to share and third, adequate means for sharing courseware must exist.

II. Faculty

The second fundamental assumption of this argument is that faculty are the key element to reaching the steady state twenty-five year endpoint. A corollary is that faculty perceptions of need are the source of effective demand for CBEM. Need will be identified by "CAI literate" instructors who must be able to recognize the potential of instructional computing and must be willing to try CAI. Faculty must be schooled to recognize when computing can help them do a better job and to use computing approaches effectively. This "CAI literacy" on the part of faculty will create the most effective form of demand for the allocation of resources necessary to reach the endpoint. For the present, let us reserve one component of the often used, often poorly defined term "faculty development" to describe that form of insight called CAI literacy. For the forseeable future we can assume demand will exceed the ability of local institutions to meet their own needs for superior courseware. Local institutions do not have the time, the dollars and the creative manpower necessary to publish all their own print resources, let alone CAI. This condition creates the necessity for sharing.

III. Courseware

There must be something to share. That something should be first-rate courseware of proven effectiveness. At this point in time, most of us are aware that some currently existing courseware appears to be effective, is suitable for inter-institutional sharing, and has become an integral part of the health education curricula. Why then, is the further adoption of CAI slow, especially in relation to the widespread perception of tremendous education potential with this medium? What is rate-limiting? Is it cost? Is it language incompatibility? Lack of reliability? Lack of communications transparency? Lack of publishing mechanisms? Lack of author incentives? What?

The third major assumption of this argument is that a very important rate-limiting factor in the development of this field at this point in time is a shortage of first-rate lessons of proven effectiveness. No one subdiscipline is supported by a CAI curriculum generally recognized as accurate with respect to content and educationally effective. What proportion of CAI units we are each aware of can truly be described as exceptional? This shortage limits the rate of incorporation of CBEM into the curricula. This limits the rate at which faculty can be enticed to try CAI. Let us assign a second component of the term "faculty development" to faculty as courseware author in distinction to faculty development qua instructor or creator of demand for educational materials. Creation of additional first-rate courseware and demonstration of its educational effectiveness should be a positive effector for CAI use and development in the professional peer arena.

IV. Networks

Networks must be developed to share courseware. Both hardwire and program transfer networks are necessary. Each has advantages and disadvantages. Much effort has been devoted to the educational, technical, political and economic questions concerning networking which will probably be described in depth by other participants. Suffice it to say that there are two important points to make: (1) Courseware sharing is a means to an end, not an end in itself, (2) Courseware sharing is obligatory. Networks that are reliable, transparent and economic are absolutely required to share the intellectual, scholarly products - courseware - produced by a very scarce resource, the distributed CAI faculty and associated, scarce, courseware development personnel.

What about priorities? It seems to me that the order of importance in developing computer-based education in the health professions is the order in which I have presented the elements, faculty first, courseware second, and means to share it, third. It is my concern that we have often deployed our resources according to a different scheme of priorities in which means to share is first, research and development concerning something to share is second, and research and development related to increasing faculty CAI literacy is third. What good are excellent means of sharing if the supply and quality of "shareables" is inadequate and if the demand simply never materializes.

V. Faculty Development

One can envision an administrative demand pull model for development of the field based on cost effectiveness and competition in the marketplace. Because, I think, the long-term development of this field will be better served by a faculty-based demand-pull model, I believe that faculty development is the key to it. There are a number of aspects to this problem.

Analysis of a recent study of CAI utilization in medical curricula[1] indicates that integration of CBE materials into all medical curricula across the nation has increased from essentially zero to approximately three hundred units within seven years according to a logarithmic rate law. Some individuals suggest that a plateau has been reached and the rate has slowed. Whether or not this is true, only further analysis will tell. What seems to be clear is that the rate of incorporating CAI in our respective local instructional environments is slow. Precious little effort has been devoted to developing additional insight into the factors that affect the rate of incorporation into the curriculum and into means of modifying it.

I am not aware that there has been a systematic effort to develop tactical approaches to faculty development. This should be an area of rigorous and concerted inquiry, and it should be supported at the federal level.

The faculty resource is a distributed phenomenon. CAI faculty are located in the several institutions with inadequate numbers in any one to constitute "critical mass". Faculty are also a two-fold resource. First, there is the faculty role in creating demand, Second is the faculty role in producing something to share, i.e., producing courseware. The shortage of first-rate courseware of proven effectiveness is a negative feedback loop to the development of faculty literacy.

Efforts to increase the courseware supply require three factors for success: (1) involvement of the most creative faculty scholars, (2) development of a research-derived science of courseware design, (3) an evolution of and adherence to a management model for courseware development, or production, that is cost-effective.

A rationally-derived, systematic, design science for computer-based education materials would include a complete set of rules (approaches, concepts, precepts, algorithms), in other words, a "grammar", for constructing superior CAI lessons with a very high probability of success. In limited areas, successful "patterns" or partial grammars exist or are known in a few centers but their scope and extent are not widely known because they have not been identified, analyzed and codified in the literature.

This situation is also congruent with the idea that there is no fully satisfactory, universal theory of learning and instruction. Research in education, psychology, and artificial intelligence, for example, from which grammar rules can be derived has either not been done or exists but has not been consolidated into the CAI design grammar. Involvement of creative faculty hinges upon the success with which the design science can be developed. Hence, significant inquiry must be undertaken and the only reasonable approach is to adopt a posture of active, rigorous, scientific experimentation.

In addition, there is the need to verify the content validity of courseware through "peer review" means and the educational effectiveness of such lessons through evaluation. With regard to my hierarchical ordering of these factors, I question whether first-rate peer review and first-rate educational effectiveness is possible or indeed worthwhile in the absence of an ability to produce significant amounts of first-rate courseware with a high probability of success.

A new discipline is in the process of formation. It is more than coincidental that the description of needs sounds like a description of factors necessary for development of an academic discipline. Research, systematic inquiry, peer review, etc., all are earmarks of a properly functioning academic discipline.

According to Platt[2], two elements stand out as critical for scientific disciplines in their most productive, i.e., golden ages. High productivity in advancing the state of knowledge is correlated with two factors: a disciplined literature including a system for informal and extremely rapid communication among workers in the field and inquiry by the method of strong inference. The former is developing, at least, in health-related CAI. This is an improvement over two to three years ago, but it is not yet adequate. According to the method of strong inference, questions which can be answered yes or no instead of maybe are isolated by rigorous analysis. The communications within a field are so rapid that often independent analysis yields similar or identical crucial questions which are tested simulataneously in more than one center. By the application of the scientific method the question is subjected to critical tests and the entire field can go forward based on a rapid sequential process of hypothesis proposal, hypothesis testing, followed iteratively based on proof or disproof of the preceeding hypothesis. Time for evolution of the laws that govern a field is decreased because of the simultaneous testing and validation of hypotheses. This does not now exist and must be developed. If there is to be research, there must be support. I think we still do not see clearly the appropriate bounds for research, all the potential contributors and certainly we have consistently underestimated the size of the problem and the dollars required.

VI. Role of Institutions

If the foregoing analysis is correct, institutions must be developed to provide, at the very least, an explicit model for the interrelationship of people and their roles to permit rapid communcation to develop and the method of strong inference to be practiced. The loosest of Webster's meanings is taken: an institution is an organized group in society, presumably focused around common goals. In this section I would like to outline briefly the roles that might be played in 1985 by one such institution, the Health Education Network, Inc. (HEN).

The major charge of users of the Lister Hill Experimental CAI Network to the Steering Committee of the Health Education Network Users Group (HENUG) of ADCIS and subsequently to the Health Education Network, Inc. was to maintain the network as a user supported and user directed institution after federal operational support ceased. This has consumed most of HENUG and HEN effort although from the beginning, the goals of the network have been much more global. The following goals written two years ago are nothing less than a prescription for development of an academic discipline:

1. To maintain and enhance medical education through inter-institutional sharing of CAI material.
2. To investigate academic and professional aspects of computer-assisted instruction, such as faculty development, curriculum development, authorship, evaluation and peer review.
3. To investigate functional and technical aspects of computer-assisted instruction such as networking alternatives, program transfer, documentation, distribution and cost effectiveness.
4. To develop computer-assisted instruction into a disciplined literature.
5. To communicate with users and medical educators in general on the status, availability and effectiveness of computer-assisted instruction.
6. To provide meetings for the reading and discussion of papers on computer-assisted instruction, to discuss policy and disseminate knowledge.

7. To extend the means through which health professionals may continue their education.

In 1985, the HEN operates via a professional administrative staff responsible to the user-elected Board of Directors via a chief executive appointed by the Board. The Network arena is divided into a Professional Network and an Operating Network. The latter has two parts: (1) a hardwire multi-communications/ multi-host network providing national user access to courseware resident in one host, (2) a Program Transfer Network providing local and regional access to courseware resident in more than one host. Decisions on which network supports particular courseware are multifactorial. To facilitate communcations and deployment of resources, professional staff will support an explicit hierarchical, resource-chaining network of people from the usership of these physical networks. This explicit structure (model) would be organized by region and level and would identify people and capabilities at each node in the network. Better communication and more efficient use of resources would result.

The Professional Network is composed of the editorial board and the Division of CBE Development. An elected, staff-supported, discipline-oriented editorial board derived from the academic associations and professional boards conducts and publishes courseware appraisal (the CBE analogue of peer review) and certifies materials for distribution/publication on the Operating Networks. The Division of CBE Development assists by facilitating educational effectiveness research. This group also collaborates with users in the conduct of workshops,in faculty development, in analysis of need for courseware development, and in serving as a resource to courseware development teams in member institutions, etc. Institutions can get consultations in establishing local policies for copyright, royalties and author incentives.

Why is the HEN involved in so many areas? Simple. Everything is related to everything else. For example, development of consensus within the network on appropriate standards for publication requires widespread input and certification of courseware for publication requires several kinds of judgements including application of content standards by the editorial board, the education standard by the Division of CBE Development and the programming and documentation standards by the Operating Networks. These activities are synergistic and symbiotic. In analogy to biology, coordinated (balanced) development is viable and yields growth; uncoordinated development is often defective or not viable. Optimal growth requires close coordination and that requires good nervous and endocrine systems, both of which work best within a single organism. Hence, the postulate of broad-scale HEN activity and a unitary governing structure.

Submitted by:

Robert G. Votaw, Ph.D.

[1] C.S. Tidball. Personal communication.

[2] J.R. Platt. "Strong Inference," Science 146: 347-353, 1964.

Possible Future for Courseware Sharing

by

Thomas Chen and Vincent H. Wu
School of Basic Medical Sciences
University of Illinois at Urbana-Champaign
College of Medicine

The problems associated with courseware sharing may be summarized as:

1. Coursewares are developed in different languages while there exists no single system that supports most if not all of these languages.

2. Lesson presentation follows different conventions and formats which results in user frustration.

3. System limitations may hinder the man-machine interaction and the execution of complex lessons.

4. CAI materials are developed according to local curriculum which may not be appropriate for another institution.

We propose that the first step in attacking the above problems is the development of a depository network that supports multiple computer languages and provides easy user interface in lesson development/presentation, and in effective inter-user communication.

Such a depository network will consist of two types of nodes: the depository node and the local node. The depository node is an information center through which the local nodes deposit, retrieve and transfer information. The local node provides multiple CAI languages and supports the man-machine interfaces to users. To achieve a desirable cost:benefit ratio from a local node, a group of users must share one system. However, this system should be small enough so that local control, both administrative and technical, is possible. The major elements of information exchanges between the local node and the depository node are lesson materials, notefiles, and inter-user communication.

The flexibility of the network and the low start-up cost of a local node ($200,000 for a 20 terminal system, excluding terminal cost) permit the establishment of a distributed computer network in an incremental fashion. The development of lesson format standards and materials independent of local curriculum will occur when many institutions are connected together as local nodes in a depository network. This provides a convenient means for the sharing of lesson materials, and the support of online inter-user communication.

A minicomputer-based CAI system which supports Fortran, BASIC, PASCAL, PILOT, LISP, and a major subset of TUTOR has been operational for the past year and is described in a previous section of this conference. This system is a local node prototype, and it presently supports different man-machine interfaces such as plasma, CRT, and teletype terminals. Several lessons have been converted from the PLATO system to the small system in less than 5% of the original lesson development time, and with comparable performance.

Sharing Courseware

Alfred Bork
Computer Development Project
University of California
Irvine, California 92717

So far remarkably little sharing of nontrivial computer courseware has occurred.

Perhaps the main attempt at widespread sharing has been that pursued by CONDUIT. Relatively straightforward modules have been reviewed and certified, both for competence in the subject area and transportability. Because of the importance of transportability, in an environment where there are many different computers, these materials tend to be the simpler materials. CONDUIT surveys for the recent state-of-the-art report indicate that most people still use their own material, and there is relatively little use of materials available from elsewhere.

Sharing of courseware has a variety of different aspects associated with it, a number of dimensions must be considered. The first, and most often mentioned, is simply the technical problem of whether materials developed on machine A easily can be made to run on machines B, C and D. A second dimension is the question of time. Is one talking about immediately moving things to a vast number of new installations, or is one talking about this being done some time in the future? The third aspect concerns the question of the sociological problems involved in transfer, the typical problem involved in introducing materials into any new environment.

Related to all this is the question of when we expect very widespread use of computers in education to take place. My own feeling is the estimates in the Carnegie report, The Fourth Revolution, are still reasonable estimates. They suggest the period of general introduction starting about 1990, and very widespread use in higher education starting in the year 2000. So mass use is still some time away.

The truth is that widespread use would be impossible under the present circumstances. There is simply not the computer capacity available for educational use at the present time that would allow widespread transportability. In our beginning physics course at Irvine in the fall quarter (1976) we used approximately 300,000 terminal minutes each month within the class, on graphic displays. The number of systems that can provide that level of access for a single class are at the present time small.

Hence, particularly for the serious developer of courseware, the long-range problem is the more interesting one, where long-range means perhaps ten or fifteen years from now. This likely time scale suggests that current efforts should go primarily into more experimentation into how to use the computer most effectively in courses.

Unlike some, I believe that we are still relative amateurs and have much to learn. Undoubtedly we have made some progress, and certainly some compelling examples of computer-based educational uses have been developed in a number of places. But much of the existing computer-based educational material is only kindergarten material, early bumbling attempts to learn how to use a new and extremely powerful teaching medium.

Often too the limitations of computer systems impose completely unacceptable pedagogical restrictions. Thus, it seems to me that any current nongraphical materials are almost certain to be abandoned in any future system; the role of pictorial information is so important in education in all levels and in all areas I cannot conceive of it being left out.

Often the question of transportability is equated to the development of a suitable language, and then persuading everyone to use it. An example of a project based on this philosophy was the development of PLANET. At this time, surveying the number of languages that have been developed with transportability in mind, it would seem that almost anyone would conclude from the evidence that something is wrong with that approach! Most of these languages have been done in an inflexible way, so that as new modes of educational use came the languages either had to be greatly supplemented by outside materials; thus they lose their identity almost entirely, or must be abandoned. Relatively few powerful language systems exist, so perhaps we are getting to a language almost by default. To reiterate, the creation of a CAI language does not solve the problems of transportability. Indeed attempts to write in such a way that material will run on a great many different computers often lead to serious compromises in the pedagogical quality of the materials.

Along with this over-emphasis on authoring languages there is an under-emphasis on authoring systems. The entire process of creating computer-based educational materials is much more complex than simply providing an authoring language, as many computer center directors have discovered when they find that the authoring language, so proudly acquired, sees little usage. The authoring process needs to be thought through fully, rather than simply left as a by-product of the language. If we are to move away from our cottage industry stance, and get into more professional development of courseware, we must worry about the conditions under which this development can occur--the organizational structure, the intended audience, the marketing aspects.

The general problem of marketing courseware has received very little attention, with the exception of CDC's recent PLATO efforts. Yet this is a critical dimension. Little of the marketing mechanism for books can be found for courseware; for example, very little computer courseware is now paying royalties to the author. The difficulties of introducing new educational methods are well known, and deserve continued research efforts.

We can finally ask what kind of hardware is likely to lead to widespread courseware sharing. The thrust of LSI technology implies that the stand-alone systems are likely to be the dominant educational system of the future, with some interconnection with other machines for such purposes as access to large databases or extensive computational capability. I should hasten to say that I have in mind much more powerful stand-alones than current ones. Furthermore, I would expect the stand-alones to have much more elaborate graphic capability than any present systems. They should have full color and real animation, for example, and they should offer a full multimedia environment, including video, sound, and slides.

Another educational development at the present time fits in very well as a component of a stand-alone educational computer system, the home videodisc system. These systems have not been developed for this market, but for the home entertainment market. Yet they provide extremely cheap read-only storage.

The videodisc provides a sellable product. It would be the method of sharing, and more importantly marketing the computer-based educational courseware of the future. An individual disc would contain a full sequence of learning materials, including the computer code, slides, video sequences, and audio sequences. All of the media would be under the control of the computer program. A variety of such discs would be possible. Some of them might simply contain a full language, along with perhaps a set of learning materials for students learning how to use that language. Another disc might contain a large group of art slides, with a convenient indexing system allowing immediate access to any of them. Another disc might contain an elaborate CAI learning program, with or without audio-visual aids as deemed necessary by the individual instructor. Prospects are that the discs will sell for about $10, so they will be relatively inexpensive. Furthermore, they will be sellable in a wide variety of stores, and even can be handled through libraries.

We expect the computer system to contain more than the videodisc storage, since this is only read-only storage. Thus, a floppy or something like that, in addition to real core, would also be present. Probably new modes of organizing the computer, with a variety of CPU chips for different purposes, would come into use. These systems might even be modular, allowing new pieces to be plugged in for programs with specialized demands.

Finally it seems clear to me that courseware will be shared, on an extremely wide scale. But I think that the time scale is such that we cannot expect immediate widespread sharing. Some sharing will occur at the present time along with continual experimentation on how to use the computer effectively in learning.

A Glimpse of the PLATO V Terminal

by Jack Stifle

Computer-based Education Research Laboratory
University of Illinois
Urbana, Illinois

Introduction

The PLATO IV system represents one of the most successful non-engineering applications of interactive computer graphics technology. From the time of its emergence about 16 years ago until the development of PLATO IV, this technology was almost the exclusive tool of computer scientists and engineers, being used primarily in engineering applications such as aircraft, automobile and electronic circuit design. Such limited use was due in part to the high cost of graphics hardware and, in part, to the esoteric nature of computer languages.

The designers of the PLATO system recognized these problems and succeeded in developing new solutions to meet them. The results were the PLATO IV terminal[1], possessing powerful graphical and alphanumerical display capability, a communication network[2] that permits a response time of less than .2 seconds, and, the TUTOR language[3], which could be understood and productively used without the need to possess a degree in computer science.

Evidence of the success of PLATO is the existence of some 1500 terminals now in operation in five systems in the United States and Canada with more being added.

What does the future hold for PLATO? The purpose of this paper is to discuss the capabilities of the new hardware now being developed for future PLATO systems. This discussion will focus on the next generation terminal where the enhancements will occur first and be most visible to the user. Another paper in this session will discuss the future software capabilities.

PLATO V Terminal Architecture

The PLATO IV terminal represented a significant breakthrough in the field of interactive computer graphics. For the first time it was possible to provide, at a reasonable cost, line graphics, selective erase, a large dynamically alterable character vocabulary, rear screen projection capability, a touch sensitive display, and the ability to operate peripheral devices in a single terminal. These PLATO terminals have been described as "-perhaps the most sophisticated devices ever developed for communicating with a computer."[4] By taking advantage of recent developments in the field of large scale integrated circuit(LSI), principally the microprocessor and semiconductor memories, it is now possible to add even more performance capability to the PLATO terminal and, at the same time, reduce the cost.

The PLATO V terminal architecture is actually a micro computer system which contains all of the standard features of large computer system including a processing unit, a memory, and an input-output(IO) structure. This system is made to perform as a PLATO terminal by providing a plasma display panel as part of the IO structure.

As in any computer system, the contents of the memory specify the operations to be performed and, in this case, they define the operation characteristics of the terminal. It is this programmable characteristic feature which enables the same terminal architecture to perform as a PLATO IV terminal or as a more powerful terminal with enhanced capabilities.

A block diagram depicting the main functional units of the terminal is shown in Figure 1.0. Operation of the terminal is under the supervision and control of an 8080A microprocessor constructed on a single LSI chip. This chip is an 8-bit parallel processor with a basic 2us instruction cycle time and contains six 8-bit registers, an 8-bit accumlator, four flag bits, a 16-bit program counter and a 16-bit stack pointer register which is used to control a last in-first out stack contained in memory.

An 8-bit bidirectional data bus connects the 8080 to the other functional units within the terminal and to IO devices external to the terminal. The data bus acts as a highway over which information flows between internal(and external) portions of the terminal. In addition to performing display generation functions, the 8080 must manage the flow of information on this highway.

The terminal memory consists of up to 8k(k=1024) words(8 bits) of read only memory(ROM) and 8k words of random access memory(RAM). The lower portion of ROM contains the resident program and the upper 2k contains the data for the standard PLATO charcter set. More will be said later about the resident program and RAM memory.

The Serial IO port provides the link to the PLATO central computer facility.

The Display Interface Unit(DIU) contains the registers and control circuits required to efficiently attach a plasma panel to the data bus.

The Interrupt Control Unit(ICU) provides the means of initiating a request for service to the microprocessor.

The IO Buffer provides for the attachment of external devices to the data bus.

Description of Operation

A device requesting service originates an interrupt and presents it to the ICU. Within the ICU the interrupt requests have a wired in priority, and the ICU will pass to the 8080 the interrupt request having the highest priority.

When an interrupt request is accepted by the 8080, the program presently in process is halted and program control is transferred to the processing routine for the interrupting source. For example, if the terminal were in the process of drawing a line on the plasma panel and the user pressed a key on the keyboard, the line drawing process would be temporarily suspended, the key would be read by the microprocessor and sent to the serial IO port for transmission to the central computer. Following this action the line drawing process would resume. All of this activity would take place without the user observing any delay in the line drawing operation.

Eight interrupt sources are provided and they are described below

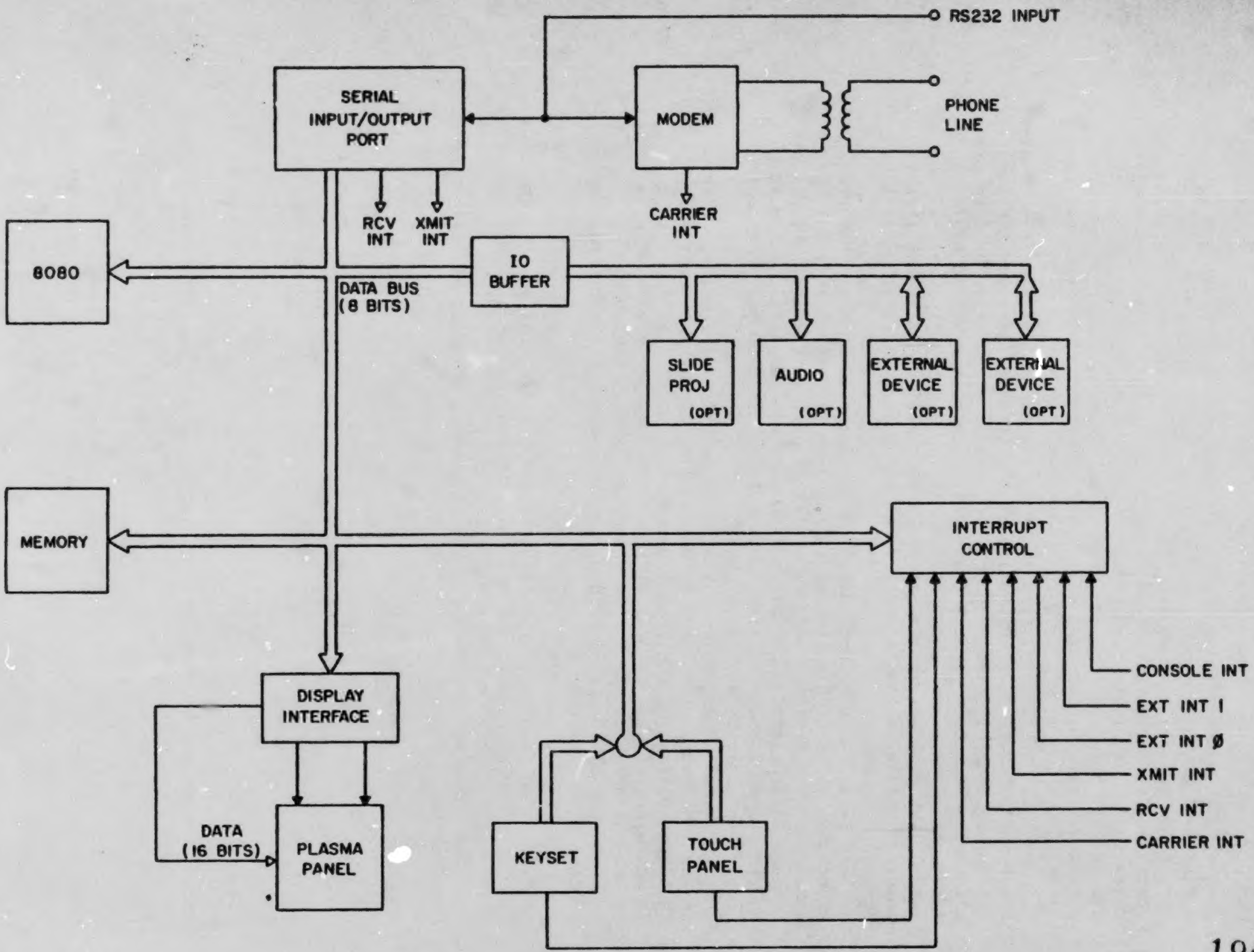

RS232 INPUT
SERIAL INPUT/OUTPUT PORT
MODEM
PHONE LINE
RCV INT
XMIT INT
CARRIER INT
8080
IO BUFFER
DATA BUS (8 BITS)
SLIDE PROJ (OPT)
AUDIO (OPT)
EXTERNAL DEVICE (OPT)
EXTERNAL DEVICE (OPT)
MEMORY
INTERRUPT CONTROL
CONSOLE INT
EXT INT 1
EXT INT Ø
XMIT INT
RCV INT
CARRIER INT
DISPLAY INTERFACE
DATA (16 BITS)
PLASMA PANEL
KEYSET
TOUCH PANEL

Figure 1.0. Terminal Block Diagram

in the order of decreasing priority.

The Receive(RCV) interrupt indicates the arrival of a new byte (8 bits) of PLATO data in the Serial IO port. The 8080 responds to this interrupt by reading the data. After three bytes have been received the data is reassembled into a PLATO work and processed.(Since a PLATO word is 21 bits in length, the extra bits are retained and used in the assembly of the succeeding word.)

The Keyset(KST) interrupt indicates that a key has been pressed on the keyboard. The 8080 responds to this interrupt by reading the key.

The Touch Panel(TP) interrupt indicates that a touch panel input has occurred. The 8080 responds to this interrupt by reading the touch address.

The Transmit(XMIT) indicates that the Serial IO port is available for transmitting data to the central computer. The 8080 responds to this interrupt by sending data to the Serial IO port if data is awaiting transmission. Otherwise, this interrupt will be disabled.

The External(EXT0 and EXT 1) interrupt indicate that some device attached to the external bus has requested service. The type of service requested will be a function of the device.

The Console interrupt indicates that the Run-Console switch on the front of the terminal has been placed in the console position. The 8080 responds to this interrupt by transferring program control a special program contained in the resident which permits the keyboard and plasma panel to be used as program debugging aids. In Console mode the user may display the contents of the 8080 registers, the contents of memory, alter the contents of memory, step through programs one instruction at a time, set a breakpoint or jump to some program.

The Carrier interrupt indicates that an interruption of the communication link to the central computer has occurred. The resident program contains a warning message which will be flashed on the screen in response to this interrupt.

Resident Program

The resident program contains the programs required to process PLATO data plus routines for servicing the Serial IO port, the external data bus, the Keyset, the Touch Panel and for communicating with the Display Interface Unit. In addition to PLATO IV processing ability, the resident provides the following new features:

An expanded alphanumeric capability including the ability to plot characters right to left and vertically in both directions, the ability to generate sized(boldface) characters, automatic setting of margins, an expanded character store capable of holding up to eight character sets, and a character plotting rate of 2000 characters per second.

A high speed blockerase capability which permits selected areas of the display screen to be enclosed without disturbing other areas. This function can be performed at the rate of 6000 characters per second.

The ability to generate an audio tone. This feature can be used to signal a term TALK request.

The ability to transfer program control to a program stored in RAM. These local programs can be loaded at condense time in the same manner as character set data is now loaded in PLATO IV terminals.

It is the last item above which represents the most significant new feature of the PLATO V terminal. Instead of the ability to store only data as in the PLATO IV terminal, programs or special alogrithums may now be loaded and executed locally. The data processed by such programs might originate in PLATO or in an external device attached to the terminal. When operating under local program control, the resident simply passes incoming data to the appropriate program for processing.

This programmable feature may be used to store special programs which the user could obtain from a library provided by PLATO. Such programs might include animation routines for generating geometric figures such as circles, special graph plotting routines or programs to operate peripheral devices attached to the terminal such as a printer or floppy disk.

It will even be possible to time share the plasma panel, displaying simultaneously data generated by PLATO and by a local program.

The PLATO user community has demonstrated an outstanding ability to create clever and innovative displays on the PLATO IV terminal, many of which exploit features of the terminal in ways not even remotely conceived of when the terminal was designed. This experience has proven that given adequate hardware and a natural computer language, one does not have to be a computer scientist to make productive use of computer graphics technology. It is therefore impossible to predict all the ways in which this new terminal will be utilized, but based on the experience of PLATO IV it is reasonable to expect to see significantly more dramatic and sophisticated displays.

References

1. "The PLATO IV Student Terminal", J. Stifle, Proceedings of the Society for Information Display, 13, 35(1972).

2. "The PLATO IV Communications System", B.A. Sherwood and J. Stifle, CERL Report X-44, Computer-based Education Research Laboratory, University of Illinois, Urbana, Illinois 61801.

3. "The TUTOR Language", B.A. Sherwood, Computer-based Education Research Laboratory, University of Illinois, Urbana, Illinois 61801.

4. "Computer-Assisted Instruction: Two Major Demonstrations", Science, Volume 176, June 9, 1972, pages 1110-1111.

Factors Contributing to Success in Author Training[1]

Larry Francis

Military Training Centers Project

Computer-based Education Research Laboratory

University of Illinois, Urbana-Champaign

The Military Training Centers (MTC) Group created and taught the first formal author training course for the TUTOR programming language and the use of the PLATO system. The course was used over a period of three years to train approximately 100 authors who spent two to three weeks at the Computer-based Education Research Laboratory (CERL) of the University of Illinois at Urbana-Champaign. To date, these authors have prepared lesson materials conservatively estimated at 600 lessons or 300 student contact hours (based on actual use data automatically recorded for each lesson). This paper relates the highlights of a recently published report describing the experience gained and observations made by the MTC Group (Francis, 1976).

In general, the new authors had little previous experience with computers or programmed instruction; most were classroom instructors. After they returned to their home sites to write computer-based education (CBE) lessons, members of the MTC staff provided consultation, programming support, and liaison by means of on-line communications and on-site visits. The data for this paper were gathered while administering these services to the sites.

The Course

Although topics such as instructional design and computer science were included in the author training course, most authors arriving at CERL viewed learning how to program in TUTOR as their only goal. Thirteen cognitive and affective principles guided the creation of the author course; five wer posited originally, and the rest were derived from experience gained in teaching the course. For example, the MTC staff realized that virtually none of the authors who were writing lessons on the PLATO system had ever been required to earn grades or credits by taking classes taught via PLATO lessons. Therefore, the training course attempted to give the authors the experience of receiving instruction from a computer so that they would be sympathetic to the students' needs. The developers also found that because the profession of authoring computer-based instruction lessons is a new one, many authors lacked a knowledge or sense of what was expected of them. They needed a model to suggest what behaviors were typical and appropriate in a variety of new situations. Without such a model, some chose inappropriate behaviors as exemplars (e.g., several authors felt that to be a "good" author, one must spend 100% of his time at a computer terminal).

As successive author groups received instruction, the TUTOR training portion of the course underwent major reorganization and reordering. The final organization of the course placed comparatively easy material at the end of the course. This sequencing significantly improved new authors' attitudes towards beginning their first lesson and continuing to learn more about programming. Authors felt they "built up momentum" as they finished the course.

A particular difficulty was found in trying to teach the order of execution when it was governed by the sequence of the commands, rather than by explicit branching statements. In TUTOR some commands have a secondary, implied-branching effect in addition to their more obvious function (e.g., during the process of judging a student response, certain commands are not executed in some positions but are executed in other positions depending on whether the response has been recognized). Thus, various invisible "if-then-else" structures are automatically generated by the author as he rearranges his commands. These implied structures make programming very convenient and efficient for experienced authors, but are rather difficult to explain to new authors. Some authors were convinced that a sophisticated computer such as the PLATO system wouldn't require the commands to be in a particular order. They felt that the computer could decide the proper order of execution irrespective of the sequence they had used!

Because author backgrounds varied enormously, some authors had not accomplished the training course objectives after the formal training period or even after as long as two years on the job. In some groups, authors seldom took time to read manuals or reference material explaining the system. Deficiencies in programming skills as well as others in instructional design, were recognized early. In addition, many new features were being added to the evolving TUTOR language and there were some topics which could be more easily explained after the new authors had gained experience in working on their own writing lessons. For these reasons, on-site advanced training (often including remedial components) was provided to new sites during their first year.

Advanced Training

MTC experienced a significant problem in ensuring that materials designed for self study actually found their way into the hands of the intended recipients, rather than being placed in an administrator's files. Therefore some advanced training materials were set up in a workshop or seminar format. Nevertheless, MTC staff noted a reluctance to "halt" courseware production for any sort of formal training.

An attempt to upgrade authors' calculation abilities by providing them with extremely well-documented subroutines (with obvious, wide application) was of limited success. An evaluation by potential users suggested that all authors thought that documented exemplars of coding were useful--to someone else.

Dilemmas

Several training dilemmas arose during the delivery of basic and advanced TUTOR training. Very often they arose because site management staff were not sufficiently acquainted with the procedures for generating instructional materials or operating a computer-based instruction center. For example, MTC had difficulty demonstrating a need for documentation and courseware engineering because the results of poor habits in these areas are problems only in the long term or only for persons other than the authors of the lessons. An experienced manager would presumably be well aware of the benefits of sound documentation habits and could enforce appropriate guidelines. However, such staff are difficult to find.

Another training dilemma arose because of the style of critiques requested by authors. As noted above, after their CERL training authors generally preferred to avoid further formal instruction and engage only in lesson development. Therefore, upon request, MTC examined the instructional design and programming of new lessons and prepared a critique for the author. MTC viewed these critiques as training devices for upgrading author skills; therefore, we hoped that in the long run we would be able to substitute summative comments for specific comments. Unfortunately, authors were sometimes unable or unwilling to generalize the specific comments in MTC critiques to similar cases in other lessons.

We found that the vast majority of authors experienced high ego involvement with the lessons they created. This led to a variety of problems. For example, this phenomenon manifested itself as an unwillingness by authors to discard anything they had ever written, whether good or bad, whether lesson material or only practice exercises. The MTC staff explained to the first training group that their initial lesson would no doubt be eventually discarded, as one might discard the first item in a manufacturing run. We therefore advised the authors to complete the first lesson as soon as possible and then begin a second lesson. In this case, the strategy backfired. The authors felt that, armed with the knowledge about what normally happens to the first lesson, they could beat the odds, and not "waste" the first lesson. Instead, of course, they wasted a great deal of time trying to polish a roughhewn lesson.

Promising solutions for many of these dilemmas have been posited, and limited experience in implementing them is available in some cases. For example, the task of completing or repairing a "half-baked" or "mangled" lesson has been successfully used to sharpen instructional design sensitivities. It also holds promise for raising issues that the new author would not ordinarily grapple with early in his career (e.g., documentation). Furthermore, this "puzzle solving" nature of these exercises seems to be self-motivating for many authors.

MTC staff judged that training new authors during a period of rapid system software evolution did not place undue constraints on the training staff or the authors being trained. In addition, the modifications which occurred allowed one entire topic (using the TUTOR editing features) to be dropped, and permitted other topics to be self-taught or taught in an abbreviated form (e.g., display creation).

Evaluation

The evaluation of the MTC training course was made by examining reports of users outside of MTC. The adoption of the training package by outside groups and the opinions they expressed about it suggest they found the materials valuable and effective. For example, a group of community college authors was taught the fundamentals of TUTOR using a shortened version of the MTC course. Previous authors had received instruction from the Community College Coordinator who was a senior educator with instructional design and computer-based education experience. Evaluatiors of the community college program quoted an author who summarized the feelings of community college instructors as:

> The general consensus was that the [semester-long] spring course was far superior to the fall course in every respect. One reason given was that early in the course the class spent a weekend at CERL with full access to the terminals and met with the subject matter area coordinators. They returned from that weekend with a good up-to-date understanding of the newest TUTOR commands and much practical experience. "Our group had better training. We learned the powerful commands--mode erase, variables etc." (House & Gjerde, 1973, pp. 35-36.)

The opinion cited above is consistent with the results found in an evaluation of 20 of the participants at the end of the workshop. When asked if they would recommend such a session to a friend, 12 responded "Definitely Yes"; 6, "Probably"; 1, "Uncertain"; and 1 failed to respond. None indicated "Probably Not" or "Definitely Not" (Avner, 1973).

Recommendation

The experiences from this author training course point out the need for (a) better author selection procedures, (b) improved training for managers of CBE projects, and (c) more powerful techniques for inducing authors to adopt standard, state-of-the-art procedures for efficient lesson production.

---

[1] This research was supported by the Advanced Research Projects Agency of the Department of Defense under U.S. Army Contract DAHC-15-73-C-0077 and indirectly by the National Science Foundation (USNSF C-723). Requests for reprints and correspondence should be sent to Larry Francis, Computer-based Education Research Laboratory, 252 Engineering Research Laboratory, Urbana, Illinois 61801.

References

Avner, R. A. PLATO evaluation report--TUTOR workshop 26-28 January 1973. Urbana, Ill.: University of Illinois at Urbana-Champaign, Computer-based Education Research Laboratory, 1973.

Francis, L. The TUTOR training course: Lessons learned. Urbana, Ill.: University of Illinois at Urbana-Champaign, Computer-based Education Research Laboratory, 1976.

House, E. R. & Gjerde, C. L. PLATO comes to the community college. Champaign, Ill.: University of Illinois at Urbana-Champaign, Center for Instructional Research and Curriculum Evaluation, 1973.

Internal Evaluation of the PLATO System

R. A. Avner
Computer-based Education Research Laboratory
University of Illinois at Urbana-Champaign
Urbana, IL 61801

Evaluation may be considered as a process of gathering data for people who must make decisions. Some of the data summarized here were collected as part of the ongoing evolutionary design of the PLATO CBE (Computer Based Education) system. However, they also have value for the person seeking information on the status of the PLATO system.

Accessibility

Successful implementation of educational technology is based on a three-step hierarchy of necessary-but-not-sufficient conditions. The first condition is accessibility to the technology. At the very least, accessibility implies that the hardware or software comprising the technology actually exists. A sales brochure or grant proposal may be exciting and even informative, but it is not a substitute for the product it describes. Accessibility also connotes matters such as reliability and performance.

Four PLATO CBE systems are now in operation at sites in North America. Many new installations are now being planned by the Control Data Corporation, the commercial vendor for PLATO. The data described here are for the installation at the University of Illinois. This installation is where most new hardware and software features are "debugged" and thus provides a worst-case example of operating reliability. The University of Illinois PLATO IV system has been in stable operation since 1972. It now supports 921 plasma panel (full graphics) terminals augmented by 197 random-access audio units and 717 touch-sensing panels. Total usage of the system currently exceeds 1 million contact hours each year, and over 3 million contact hours had been documented by the end of 1976. Despite several major modifications of the system during this period, almost 95% of scheduled terminal time was available to the 79 organizations using this system. Reliability of individual components of the system continue to rise as experience suggests improved designs. A typical terminal in 1973 experienced some form of hardware problem every five weeks. By 1976 the typical terminal went an average of 13.24 weeks between problems, and many of these problems merely entailed minor adjustments. More important from the view of an instructor, the probability of a class-hour being interrupted fell from .14 in 1973 to .10 in 1976. A recent study by the author of instructor attitudes toward system reliability indicates that the 1976 level is near the break-point in acceptability by instructors teaching accredited courses where CBE materials are a required part of the curriculum. Reliability of the PLATO systems which are not being used for developmental testing is, of course, substantially better than this marginal level. Even so, of a group of 24 new instructors who used the University of Illinois PLATO system for a total of 822 class sessions in accredited courses, only five expressed a belief that the system was too unreliable for them to require its use by students.

The PLATO system is designed to insure that every student on the system has rapid access to a specified quantity of computer power and computer storage independent of total system load. Specifically, every student should be able to expect round-trip key-response times of less than .2 second, an average of 2000 computer instructions executed every second, and access to disk storage an average of once each minute. Because the PLATO system shares major resources among many users, peak demands for resources far above these levels should also be tolerated for brief periods. Surveys of usage records of over 27,000 PLATO users (including 22,401 students) and system functioning under various load conditions indicate that one can be 99% confident that 99% of key echo times will be less than .2 second. For students, 85.5% of records showed average computing loads of less than 2000 instructions per second, and 75% of records showed average disk storage access rates of less than one per minute. Records for many courses which used very sophisticated simulations indicated average system demands well below these design averages, and a very few courses showed averages well in excess. Thus, the PLATO system not only meets its design standards for students, but shows evidence that sophisticated materials can be supported at these levels.

User Acceptance

The second condition for successful implementation of educational technology is user acceptance. While very favorable data from user attitude and opinion surveys could be cited, the most convincing information is the fact that demand for access to the University of Illinois PLATO system exceeds the number of terminals available. Currently some 79 different organizations located throughout the United States are using the system. Users include 42 colleges and universities, 5 elementary or secondary schools, 23 government agencies, and 8 commercial concerns. With the exception of some of the elementary schools, these organizations are paying all or a substantial part of the costs of access to PLATO themselves. Included are 6 community colleges which were formerly funded as a part of the recently completed National Demonstration project of the National Science Foundation.

Educational Effectiveness

The third condition for successful implementation is a demonstration that the technology can be cost-effective. CBE is, of course, simply a medium. It can be misused or used inappropriately, as can any medium. Thus, a demonstration of effectiveness in a given situation should only be used as an indication that such effectiveness is possible--not that it is guaranteed.

Final data from the NSF National Demonstration will be reported in full by an independent evaluator, the Educational Testing Service. However, data from other projects are available. Dare (1975) reported an application in which PLATO CBE materials were compared with materials presented by other self-paced media. Costs of preparation and revision of PLATO materials were found to be about the same as or less than the costs for preparation and revision of other forms of self-paced instruction. In several different courses, students were found to attain given levels of mastery with PLATO materials significantly faster than with other self-paced materials. In this particular situation, time savings for the students were not deemed sufficient to warrant the higher presentation costs of PLATO (students were low-ranking military personnel). Grimes (1974) described use of a PLATO simulated microbiology laboratory that would not otherwise be feasible because of costs of the materials involved. Here cost-effectiveness was provided by making use of features of PLATO that no other medium could offer.

Conclusion

Continued development of the PLATO system shows a constant trend toward higher reliability (as well as lower cost). However, even for the worst-case situation of a PLATO system used for current research in hardware and software, reliability and acceptability have been found high enough to insure a very large population of satisfied users who are producing effective instruction.

References

Dare, F. C. Final Report. Evaluation of the PLATO IV System in a Military Environment, June 1975, U. S. Army Ordnance Center and School. Aberdeen Proving Ground, Maryland.

Grimes, G. M. Computer-based Education at the U. I. College of Veterinary Medicine, Illinois Veterinarian, July 1974, vol 17, 7, 10-14.

Dr. Pauline R. Jordan
Control Data CBEducation Company

THE CONTROL DATA LEARNING CENTER NETWORK

In April, 1976, William Norris, President of Control Data Corporation, announced at a New York press conference the formation of a new company to deliver PLATO instructions. This company Control Data CBEducation Co. is headed by G. G. Smith on the 16th floor of a new office complex in the renovated plaza of downtown Baltimore. The official headquarters address is 36 S. Charles Street. One component of CBEd is a national network of Learning Centers.

Types of Centers (Use of overhead projector accompanies this presentation)

1. Internal Learning Centers

   The Centers are located in Company facilities and intended primarily for employee training programs. They are managed internally under the direction of the Human Resources Development Group.

   Present locations: Minneapolis and Arden Hills, Minnesota; Toronto, Canada; Baltimore, Maryland; Rockville, Maryland; and Sunnyvale, Calif.

2. Creative Learning Centers

   PLATO terminals are used to deliver instruction to a leisure-time consumer, in addition to extensive programs by classroom teachers, as well as other media.

   Present locations: Rossville and Edina, Minnesota.

3. Control Data Institute Learning Centers

   The 13 domestic CDI's added four-terminal PLATO centers in October, 1976. Traditionally, CDI's have taught computer programming, operations, and maintenance. PLATO will be used to facilitate the delivery of present training programs as well as expand the offerings of CDI in the area of vocational, technical, and industrial training.

   Present locations: Boston, New York, Arlington, Atlanta, Cleveland, Detroit, Chicago, St. Louis, Minneapolis, Dallas, San Francisco, Los Angeles, and Anaheim.

4. External Learning Centers

   Six External Learning Centers were opened in 1976 and 19 are schedules for 1977. These Learning Centers are attractive multimedia (PLATO, Video Tape, Audio Tape, Print, Study Carrels, conference and lounge areas) located in accessible business areas. About 90% of the marketing is devoted to business and industry with the remainder devoted to the traditional educational institutions.

Present locations: Arlington, Baltimore, Cleveland, Dallas, New York, and Philadelphia.

1977 locations: Houston, St. Louis, Atlanta, Chicago (2), Greenwich, Detroit, Pittsburg, Washington, D.C., Boston, New York, Los Angeles, Milwaukee, Oakland, Seattle, Cincinnati, Rochester, Hartford, and Long Island.

These four types of Learning Centers are further classified by size, hours of operation, equipment and staff. (Slide of Learning Center Models)

To give you some idea of what the External Learning Centers are like, I would like to show you some slides of the Arlington Center where I am located. (Slides here)

At the present time, the External Learning Centers are engaged in information dissemination regarding the PLATO System through numerous demos to leaders in business and industry interested in exploring the use of CAI in training programs.

The External Learning Center product line includes published courseware, author training, programming services, courseware design, and consultation.

I encourage you to visit the Centers. For PLATO users, information is available on both the Illinois and Arden Hills system in lesson "cdcnet". I also have available for those interested in "hard copy" a publication entitled "Questions and Answers about Control Data CBEducation Company." This brochure lists the addresses and phone numbers of all Centers in the network.

The Learning Centers are only one part of the Control Data CBEducation Company which includes Courseware Development, the Publishing Company, various Sales Organizations, and Special Educational Projects. However, time does not permit a discussion of the total organization. For additional information, please do not hesitate to contact me personally. I will be happy to help you -- even if you're looking for a job!

OVERVIEW OF THE PLATO MODERN HEBREW PROJECT

John D. Eisenberg
University of Illinois *

Introduction

Modern Hebrew students at the University of Illinois are currently using two semesters of beginning grammar and vocabulary lessons on the PLATO computer system. This lesson material was developed in parallel with, and as a supplement to, a class textbook.

This paper covers some of the major aspects in the design and implementation of the PLATO material. It is intended mostly to give the reader an idea of the type of things that this project considered before and during material development. It is definitely NOT intended as a model for other projects, as all projects have different goals and needs; indeed, a successful project rarely comes about by following a set of rules by rote.

Note: This paper is written somewhat informally; the pronoun "we" is intended to refer to the Modern Hebrew Project staff.

General format of the Hebrew curriculum

Each semester's material consists of twelve to fourteen lessons, one lesson for each week of the semester. Each of these lessons covers the same material which the book covers, usually one to one and a half chapters.

A lesson is divided into three sections; two sections of grammar exercises and one test section covering all the grammar and vocabulary presented in that lesson. Each section consists of four to six exercises. Non-test exercises may be repeated as often as desired, but test exercises may be taken only once. A single section of a lesson is designed to take twenty to thirty minutes to complete.

A student may not proceed to a subsequent section or lesson without first having completed all the exercises in the preceding section. A previous lesson may be reviewed through the Master Index for that lesson; this index contains a list of all the non-test and remedial exercises.

Vocabulary games, a lesson which allows students to write comments to the teachers, and a lesson allowing students to write notes to one another are available at all times.

---

* Currently at the computing center, University of Delaware, Newark, DE 19711

Scoring and remediation

Normally, a student receives one point for each answer given correctly within a specified number of tries. If a student exceeds another specified number of tries, he is given the correct answer (and must type it in verbatim) before being allowed to proceed. This prevents a student from being stuck at a question whose answer he does not know, and also assures us that he has at least seen and reproduced the correct answer. If the score on an exercise falls below a certain level, the student is forced to do a remedial exercise (this last applies only to non-test exercises, of course). If a student repeats an exercise, the latest score received is the one which is recorded.

Student Pacing

When the subject of student pacing arose, some project members intended to match each non-test section of a lesson to one of the PLATO class sessions, which met on Tuesday and Thursday; the test could be taken any time before the next Tuesday session. It was also intended that a person not be able to fall behind; that is, lesson one would become unavailable when the Tuesday for lesson two came around. Objections were raised to that plan on the grounds that it would be unfair to students who missed class through illness, and the programming time and overhead to allow the routing scheme to account for such special cases would be too great.

Another suggestion that students not be able to work ahead past the current week's material was roundly disapproved by the other staff members. It was felt that students should be allowed to work ahead as far as they desired, or until their ignorance of the subject matter halted them [whichever came first]. The final pacing scheme which we adopted allowed faster students to go as far as they wished; students who fell behind more than three lessons were given a warning message to that effect (along with a suggestion that they talk to their teacher about it) every time they signed on.

Basic programming considerations

All programs and courseware were written with the following absolute requirements in mind:

1. All material displayed to the student should appear in Hebrew print (as in newspapers), but all typed student responses should appear in Hebrew script form.

2. All displays in Hebrew, as well as student responses, should appear on the screen from right to left.

3. The keyboard should be set up in a phonetic arrangement between the English and Hebrew letters, insofar as possible.

The script/print requirement was necessary since, except in grade school exercises, Hebrew is rarely printed. We hoped, by our decision, to familiarize students with both types of letters, enable them to distinguish one from the other easily, and help them remember the situations in which each is used.

The second point (leftwards writing) seems to go without saying once one knows that Hebrew is written from right to left. [Note that this is NOT "backwards writing"; with respect to Hebrew, it is "forwards"!] Nonetheless, at the time the Hebrew project started, leftward writing was not available at the system level, and special routines for leftward writing and response entry had to be written by the Hebrew project staff. While discussing feasibility of leftward writing at the system level, some system programmers tried to convince us that the output rate was sufficiently fast that the students would not notice the writing appear in the wrong direction, nor would it matter. [We were also told that leftward writing involved too much expenditure of time to implement, and too much overhead in execution to be done for the relatively small number of students we would have.]

Finally, we decided to make the keyboard arrangement phonetic rather than use the standard Hebrew keyboard, as we felt it would be easier to learn, and that learning key positions was not an important educational goal. Some letters had to be assigned to arbitrary keys, but our choices were, with one minor exception, reasonable.

Since it was necessary to teach the Hebrew keyboard arrangement, we decided to include this material in the four Hebrew alphabet lessons which first semester students took on PLATO before starting the grammar lessons [see figure 1]. For those students who had some knowledge of Hebrew, and had passed a proficiency test for the first semester's material (and thus already knew the alphabet], a single lesson was prepared which gave a quick review of the Hebrew alphabet and emphasized the keyboard.

No students have complained about our arrangement being too random or difficult, and some students have attained typing speeds (in Hebrew) of over 30 words per minute.

Note that these decisions require a terminal which is capable of displaying either English or Hebrew charcters in either a rightward or leftward direction, and whose keyboard may be dynamically 'altered' to allow either type of response. It is quite clear that the PLATO system is the only existing CAI system that meets these basic requirements without extensive hardware or software modifications.

The Hebrew Lesson Driver

One of the major decisions that we had to make before writing the first semester's material was whether we wanted to use a 'driver' for the lessons or not. We defined a driver as a program which would handle such details as presenting the questions, accepting the answers, keeping score, etc.; that is, the driver would take care of the 'busy work' of a lesson and allow the teachers and programmers to develop material more easily.

Our decision for the first semester was NOT to use a driver. Our reasoning was somewhat as follows: most of the drivers we had seen were very inflexible. For example, one of them allowed a question to appear in one of two places on the screen, and the scoring algorithm was fixed. Few drivers allowed complex displays to be built; others would accept only single-word or multiple-choice answers.

Although we were not sure which scoring method we wished to use, we were fairly sure that we might need different scoring methods for different types of exercises. While we did not know what type of display we would usually have, we were VERY sure that we would have to be able to design displays that caught the student's eye and took advantage of PLATO's graphic capability. We concluded that the use of any existing driver would restrict our options too much, and we did not want to trap ourselves into a mold that was too tight.

After we had written the first semester material, we found that most of our question/answer presentation and scorekeeping followed a certain pattern, even though the parameters we used were not fixed. Thus, we decided to write our own driver to suit our needs at that time, yet be flexible enough to allow expansion. All second semester material would use the driver, and first semester material would be converted to use it.

Structure of the driver

The driver is composed of a series of calls (-do- commands in PLATO terminology) to distinct units, each of which serves a separate, specific function. Thus, there is one unit which does nothing more than present an 'ok' message at the default location (at the end of the student's response); one which simply adds one to the student's score; etc.

Along with each unit is a flag indicating whether the lesson author wishes to use the default unit or provide his own unit to perform that function.

This separation of function has two benefits; first, if the only atypical feature of an exercise is its score-keeping method, the lesson author can make his own, yet still use the driver to take care of the typical work. If, on the other hand, the only typical feature is the scorekeeping, the author can bypass the driver altogether, and simply call upon the score-keeping unit to store the final results into the student's record with the assurance that this will be done in the standard fashion.

Driver options

The driver options which can be set answer the following questions about an exercise:

Are answers expected in English or in Hebrew?

Is the prompt character visible or not? [we usually chose not visible for fill-in-the-blank exercises.]

Are the questions presented in numeric order or in random order?

Does the question have a prelude (i.e., some unit to be done before the actual question is presented)?

Does the question have a postlude (i.e., some unit to be done after the question is judged correct.)? [In those exercises where the student's responses were put into a box at the top of the screen to form a coherent paragraph, the function of creating this paragraph was given as the postlude unit].

Who provides the 'ok' message -- the driver or the author?

Who provides the 'no' message -- the driver or the author?

Who provides the correct answer -- the driver (at a default location) or the author?

Who keeps and displays the score -- the driver or the author?

The driver also allows the author to set the following parameters:

How many tries does a student get before the correct answer no longer counts? [default 2]

> How many tries does a student get before being shown the correct answer? [default 3]
>
> How many problems in the exercise?
>
> What is the minimum score a student may get on the exercise and not be forced to do a remedial exercise?

Other options and parameters are used to determine which preludes, questions, and postludes are to be replotted on the screen when returning from a help section.

This driver is sufficiently versatile to drive such diverse exercises as a typical drill-and-practice paragraph comprehension exercise [figure 2], an exercise where a spinner points to the correct pronoun to use for conjugating a past-tense verb [figure 3], and a hidden-word game [figure 4]. Only three or four exercises in the two semesters of material cannot fit into the paradigm of the driver; one of these is the crossword puzzle exercise [figure 5].

Results of using the driver

Conversion of exercises from the first semester was not difficult since most of them fit into the driver paradigm (only natural, as the driver options were derived from their characteristics). Speed of development of second semester material was also greatly enhanced; we were usually able to enter and test one week's worth of material in two and a half to three weeks.

The driver also gives our material 'unity'. Students are fairly assured of seeing consistent displays for such things as running score and final score for exercises (since, in most cases, we use the default). If a single exercise fails to work properly, we first look at its parameters to see what is going wrong. If a whole series of exercises fails to work, we normally need to make repairs in only one place -- the driver.

In short, we are very pleased with the freedom that the driver has given us in terms of creating and modifying material.

Data Collection

As you have seen, we are using PLATO mostly to provide 'automated homework' with instant feedback. Thus, our data collection is oriented more towards providing teachers with a 'gradebook' record of student performance rather than data collection for statistical analysis of PLATO's effectiveness.

The teacher's data display consists basically of a list of completed exercises for a student along with scores for each exercise. Although teachers could access these scores on-line [figure 6], most of them preferred to have hard copy which they could take home with them to enter into their own gradebooks. Figure 7 shows part of a document given to teachers to help them interpret these hard-copy student score reports.

We also provided students with a lesson that allowed them to look at their own scores at any time.

Use of the computer in textbook preparation

Our project had been using the computer to merely present material and collect answers; the fact that we used it twice in helping us prepare textbook material came about somewhat by accident.

Our first non-CAI use of the computer occurred the day before we sent the first semester book to the publishers. The book already had a Hebrew-English word list in the back, but we had forgotten to make an English-Hebrew glossary. We wrote a short sorting program, entered the word list that was already done, alphabetized the English words, and got a printout of the result for the other glossary. We also used this method to create the word lists for the second semester book.

Our other use of the computer involved the creation of an on-line data base. We had previously used a set of index cards to keep a list of all the words that had been presented in the book along with such information as what lesson/exercise they were first used in, whether they had been actively taught or passively presented, what part of speech each word was, etc. One day, it simply occurred to one of our project staff that the job could be done better by computer, and, within a week, the program that kept all this information on-line was finished. Figures 8 through 10 show some of the displays from the program. One example of how this program could be used would be to find all verbs actively taught before lesson seven for use in a midterm vocabulary review exercise.

The point of all this is to realize that the computer can (and, indeed, should be expected to) do more than just CAI.

Documentation and maintenance

Documentation of the inner workings of the lesson material is, of course, necessary if any of the organizers of the project leave (and they will). Proper documentation will help those who have to maintain the courseware when the language changes (and in TUTOR, this can be an enormous problem). There is little need to go into a discussion of the value of documentation here; far better work on the subject may be found in

The Psychology of Computer Programming

by Gerald Weinberg, and

The Elements of Programming Style

by Kernighan and Plauger.

The Hebrew project's main documentation consists of a separate copy of the driver (including the leftwards writing and response routines); this copy has generous spacing and copious comments. [The only drawback with a separate copy is making sure that it is updated when the production version of the driver is changed.] Also, flowcharts of the leftward writing and response routines, as well as diagrams of how student data are stored in the student variable bank are on file with the people currently in charge of the Hebrew project.

The remainder of this paper consists, for the most part, of personal opinions that this author has formed while working on the project.

The programmer as teacher

People who are programmers by profession should not be expected to be good teachers. Most programmers, upon first learning a CAI language, tend to produce as their first efforts very well-written code for a frightfully boring page-turner (usually surrounded by fancy displays in the case of PLATO). Teaching, and knowing what type of material students like and dislike, is something of an art. One should not expect a programmer, especially one with no formal training in such matters, to be able to teach well.

In the case of the Hebrew project, I, for one, knew next to nothing about teaching and knew very little of Hebrew besides the alphabet.

In a case such as this, it is unwise to ask such a programmer to act as material developer. However, one may make use of the programmer's ignorance of the lesson material to use him/her as a 'test student'. That is, independent of programming difficulty, a programmer can make judgments about the quality of a lesson from somewhat the same standpoint of a student in the course (i.e., lessons too boring, too difficult, etc.).

The teacher as programmer

On the other side of the coin, people who teach by profession should not be expected to be good programmers. Most teachers, upon first learning a CAI language, tend to produce as their first efforts very exciting, educational lessons (usually surrounded by fancy displays in the case of PLATO), built upon a foundation of frightfully bug-ridden code. Programming, and knowing what types of code and data structures work well and do not work well, is something of an art. One should not expect a teacher, especially one with no formal training in such matters, to be able to program well.

In the case of the Hebrew project, while the teachers had an excellent knowledge of the material they wanted to teach (being native speakers of Hebrew), it is quite fair to say that none of them could have programmed the lesson driver or student router (if they had any desire to write them) in the same amount of time -- or as well as the programmer who was hired to do so.

Thus, in our project, it turned out that the teachers had little desire to learn enough programming to be able to understand (or write) the lesson driver, but were quite eager to use the driver to enter good lesson material. Conversely, the programmers were quite eager to figure out the algorithms and data structures needed for the driver, but really had very little idea of what to do (or what could be done) with the finished program.

Teacher/programmer communication

There are, of course, teachers who are excellent programmers, and programmers who are also excellent teachers; these people, however, seem to form a very rare breed. In the event that you are unable to staff your project entirely with such talented personnel, you may have to separate the teaching and programming functions.

In this case, the following simple rule will help your programmers and teachers get along with one another:

LISTEN.

Although the teacher is the last word on course content and teaching method, he should listen to the programmer who thinks that an exercise is unclear or boring. Don't forget that this view may be closer to that of a student than you think. If a programmer tells the teacher that an exercise is too expensive to implement, or is beyond his abilities, the teacher should listen; after all, the programmer knows his limits better than the teacher does.

Although the programmer has the last word on algorithms and data structures, he should listen when the teacher gives specifications for a lesson. Otherwise, the teacher may end up with what the programmer THOUGHT was desired instead of what was really desired. Listen when a teacher says that a certain software convention makes things too hard for students to use the computer; the teacher is closer to the student's viewpoint in that he may not be as familiar with using a computer as the programmer is.

The major point to remember when listening, of course, is that while the lines of communication are opened, neither party is obligated to take any of the advice they listen to.

Student reaction and input

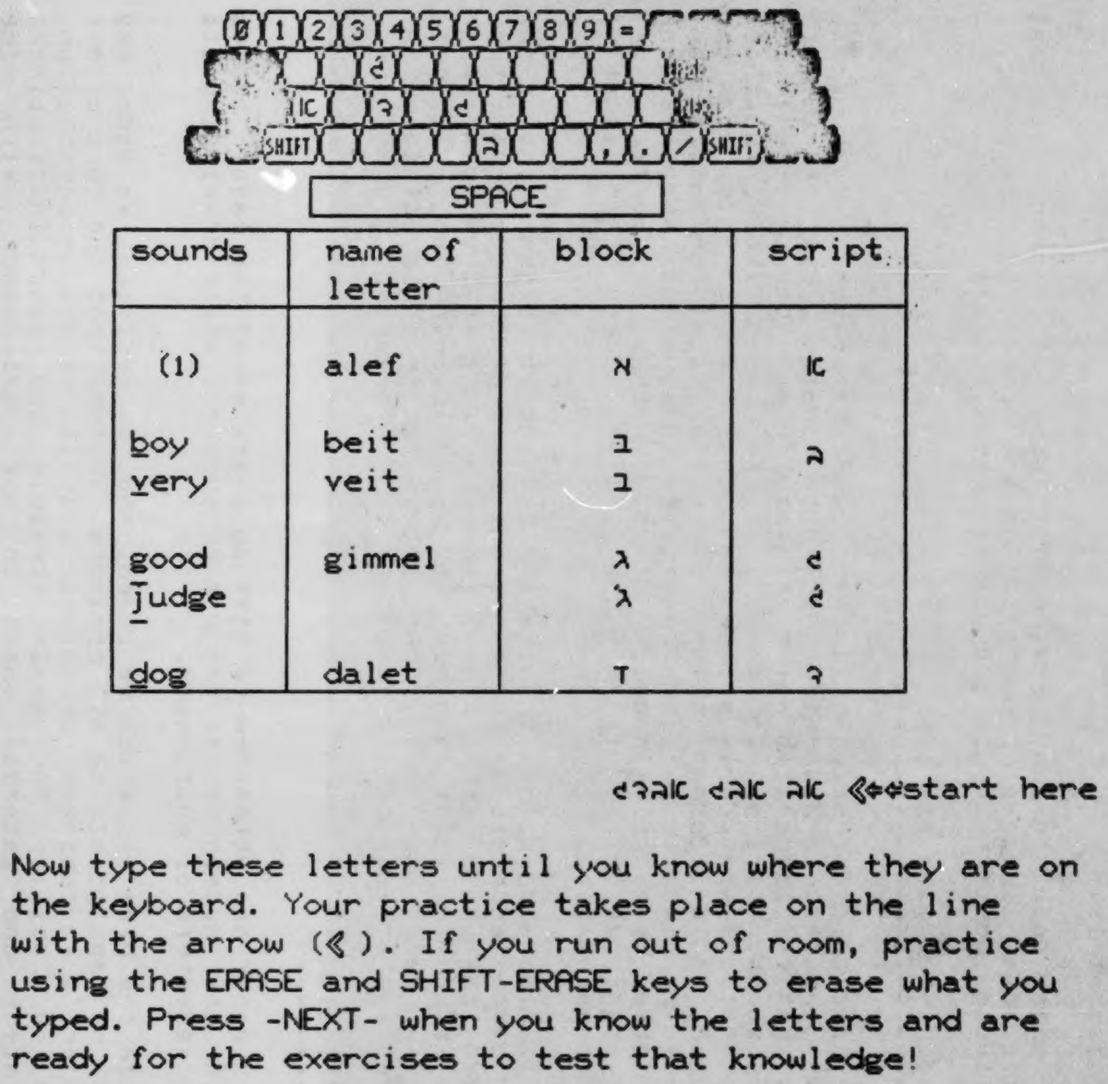
The same rule for helping teachers and programmers communicate also applies to student/project communication. If the students say that the instructions for an exercise are too difficult to understand, listen to them. It is more likely that the instructions are poorly written than that the students are too stupid to understand them; in either case, the instructions should be rewritten. [One exercise in the alphabet lessons suffered from this problem so badly that it could not be salvaged; we deleted it from the lesson rather than fight a hopeless battle.]

It is also a good idea to give the students questionnaires at the end of the semester asking them what parts of the curriculum they liked the most (so that you can do more of it), and what parts they liked the least (and how they would improve it). It lets the students know that you are interested in their opinions, and gives you an idea of how your material is accepted.

Conclusion

This paper has attempted to give an overview of the major topics that the Modern Hebrew project encountered while developing its two semesters of courseware.

While the problems which we faced, and the decisions we made, may not be applicable to all projects, we feel that the core of our project's success was our careful planning before implementation, and our ability to remain flexible and revise instructional strategy and material based on our experience with the courseware.

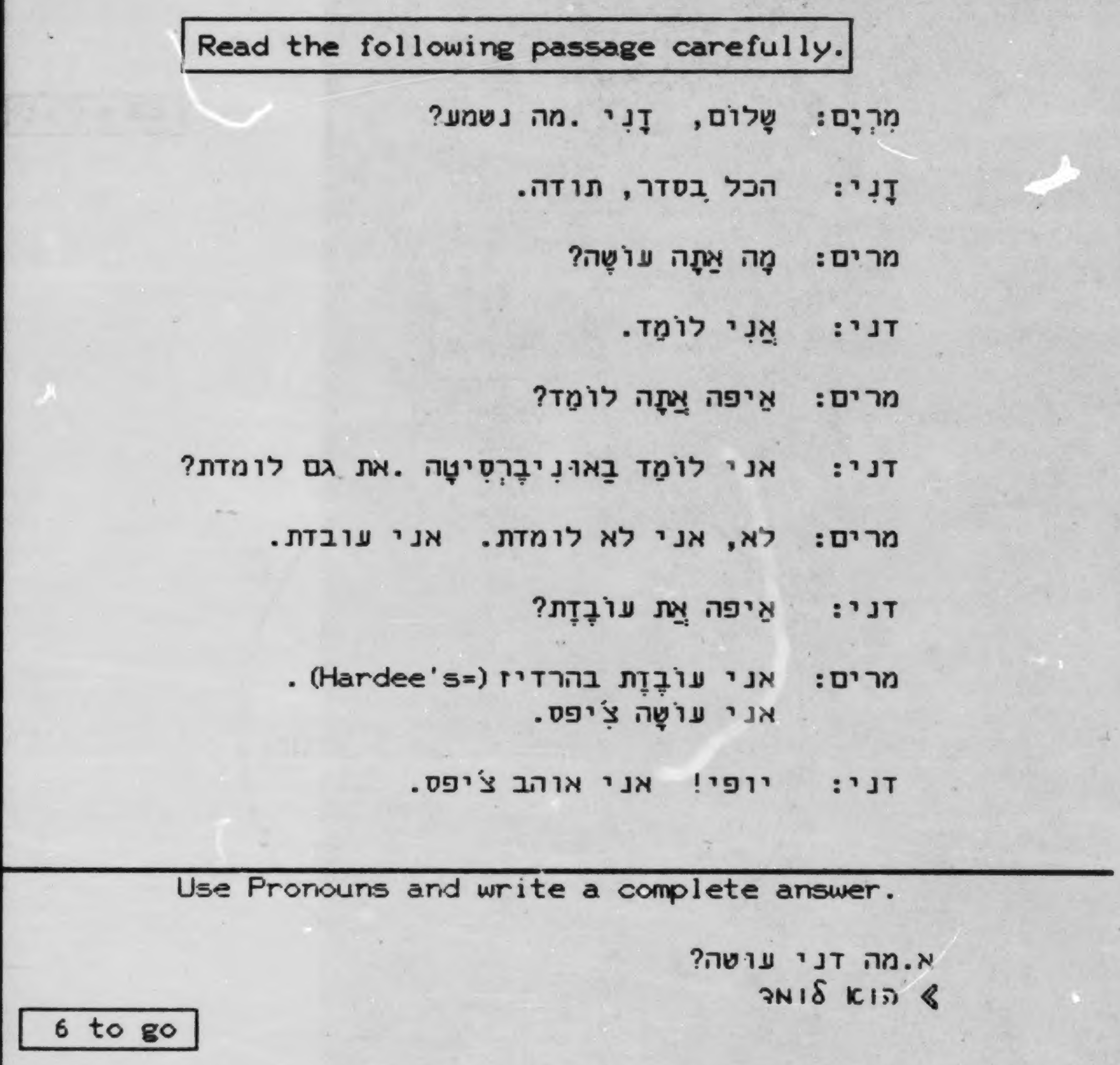


sounds	name of letter	block	script
(1)	alef	א	א
boy	beit	בּ	ב
very	veit	ב	
good	gimmel	ג	ג
judge		ג'	ג'
dog	dalet	ד	ד

אבדג אבג אב ≪⇐⇐start here

Now type these letters until you know where they are on the keyboard. Your practice takes place on the line with the arrow (≪). If you run out of room, practice using the ERASE and SHIFT-ERASE keys to erase what you typed. Press -NEXT- when you know the letters and are ready for the exercises to test that knowledge!

Figure 1



Read the following passage carefully.

מִרְיָם: שָׁלוֹם, דָנִי .מה נשמע?

דָנִי: הכל בְסדר, תודה.

מרים: מָה אַתָּה עוֹשֶׂה?

דני: אֲנִי לוֹמֵד.

מרים: אֵיפה אֲתָה לוֹמֵד?

דני: אני לוֹמֵד בָּאוּנִיבֶרְסִיטָה .את גם לומדת?

מרים: לא, אני לא לומדת. אני עובדת.

דני: אֵיפה אֲת עוֹבֶדֶת?

מרים: אני עוֹבֶדֶת בהרדיז (=Hardee's) .
אני עוֹשָׂה צִ׳יפס.

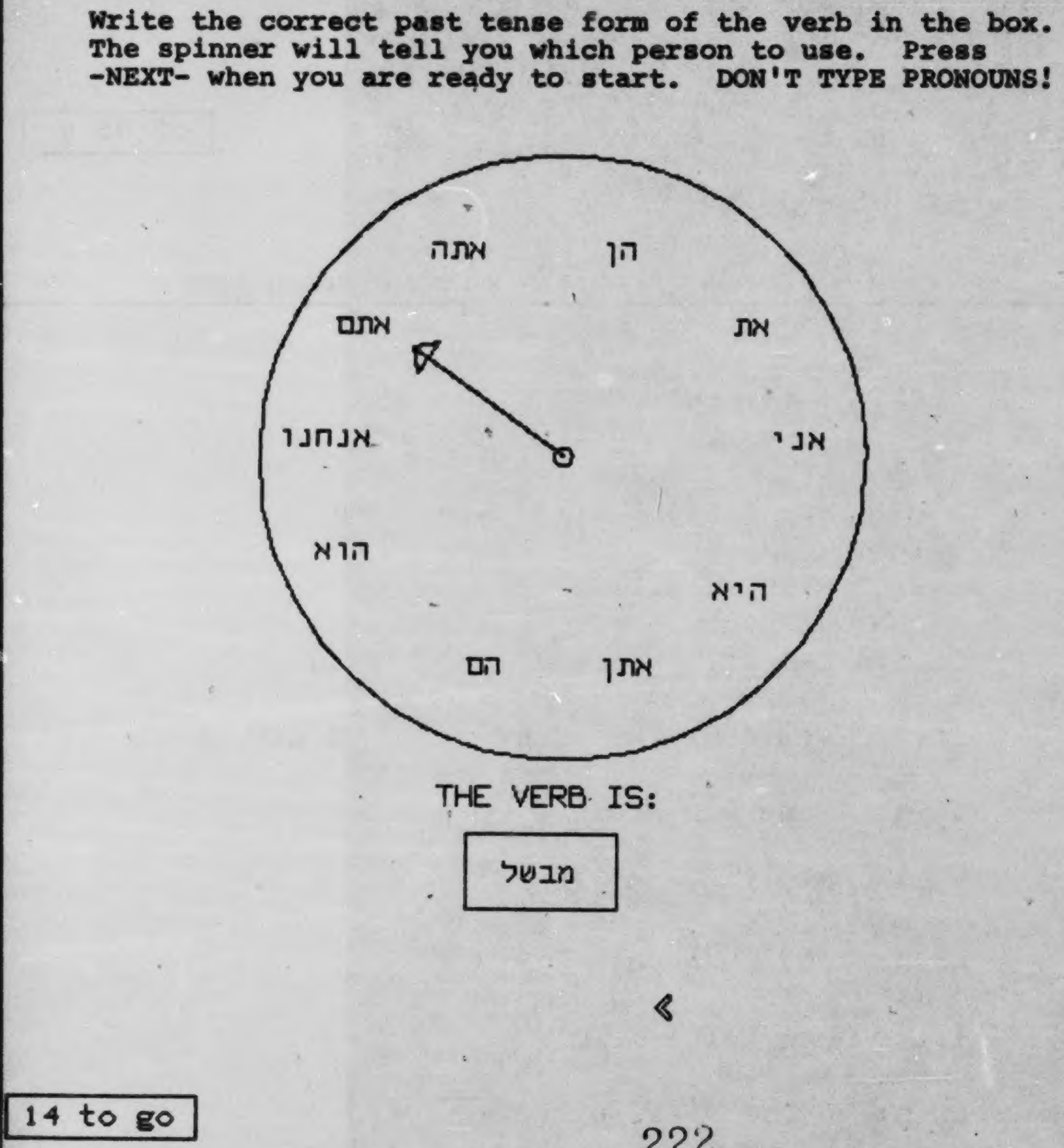
דני: יופי! אני אוהב צ׳יפס.

Use Pronouns and write a complete answer.

א.מה דני עושה?
« הוא לומד

6 to go

Figure 2



Write the correct past tense form of the verb in the box. The spinner will tell you which person to use. Press -NEXT- when you are ready to start. DON'T TYPE PRONOUNS!

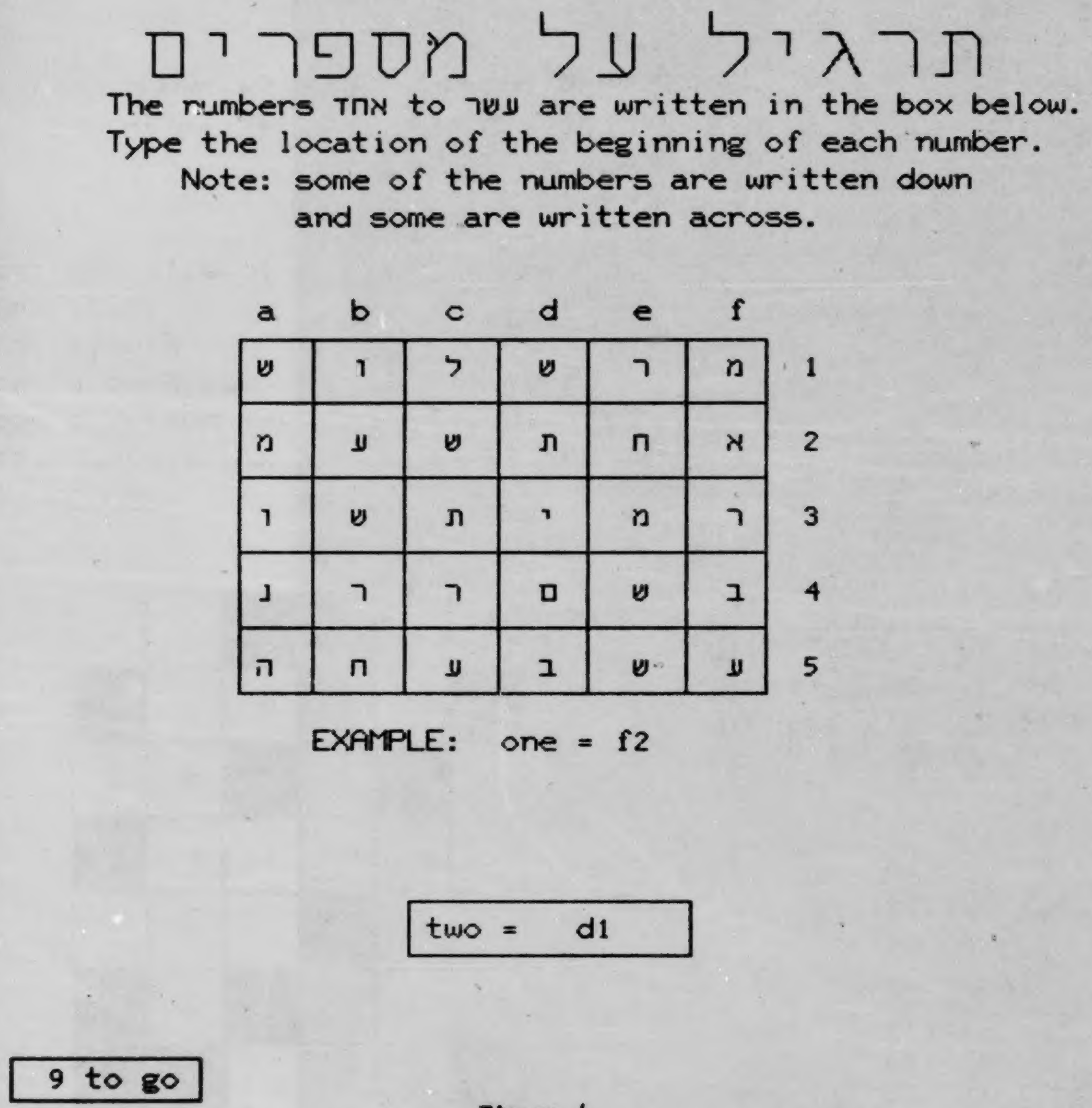
אתה
הן
אתם
את
אנחנו
אני
הוא
היא
הם
אתן

THE VERB IS:

מבשל

14 to go

Figure 3



תרגיל על מספרים

The numbers אחד to עשר are written in the box below.
Type the location of the beginning of each number.
Note: some of the numbers are written down
and some are written across.

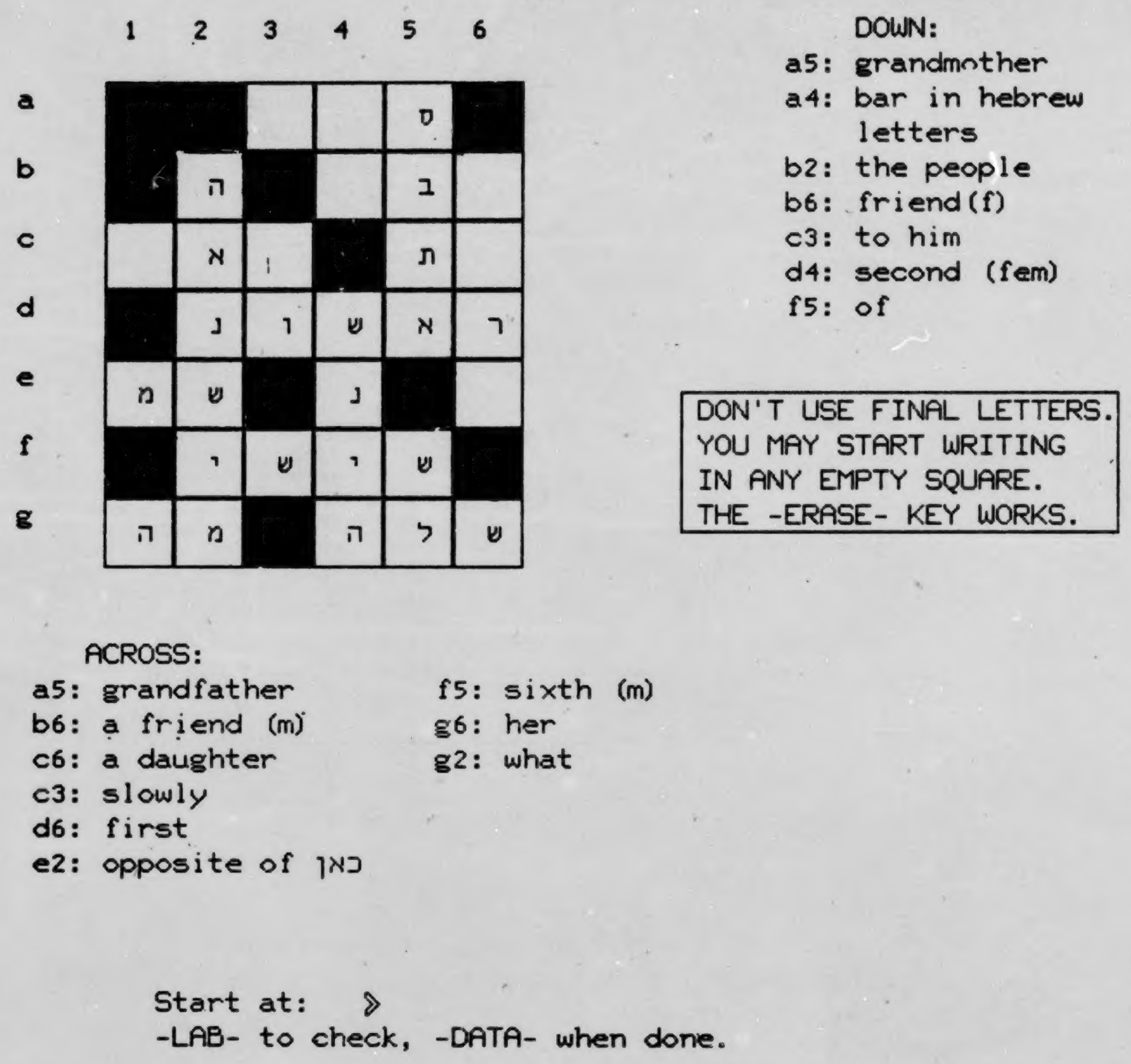
a	b	c	d	e	f	
ש	ו	ל	ש	ר	מ	1
מ	ע	ש	ת	ח	א	2
ו	ש	ת	י	מ	ר	3
נ	ר	ר	ם	ש	ב	4
ה	ח	ע	ב	ש	ע	5

EXAMPLE: one = f2

two = d1

9 to go

Figure 4



	1	2	3	4	5	6
a	■	■			ס	■
b	■	ה	■		ב	
c		א		■	ת	
d	■	נ	ו	ש	א	ר
e	מ	ש	■	נ	■	
f	■	י	ש	י	ש	■
g	ה	מ	■	ה	ל	ש

DOWN:
a5: grandmother
a4: bar in hebrew letters
b2: the people
b6: friend(f)
c3: to him
d4: second (fem)
f5: of

DON'T USE FINAL LETTERS.
YOU MAY START WRITING
IN ANY EMPTY SQUARE.
THE -ERASE- KEY WORKS.

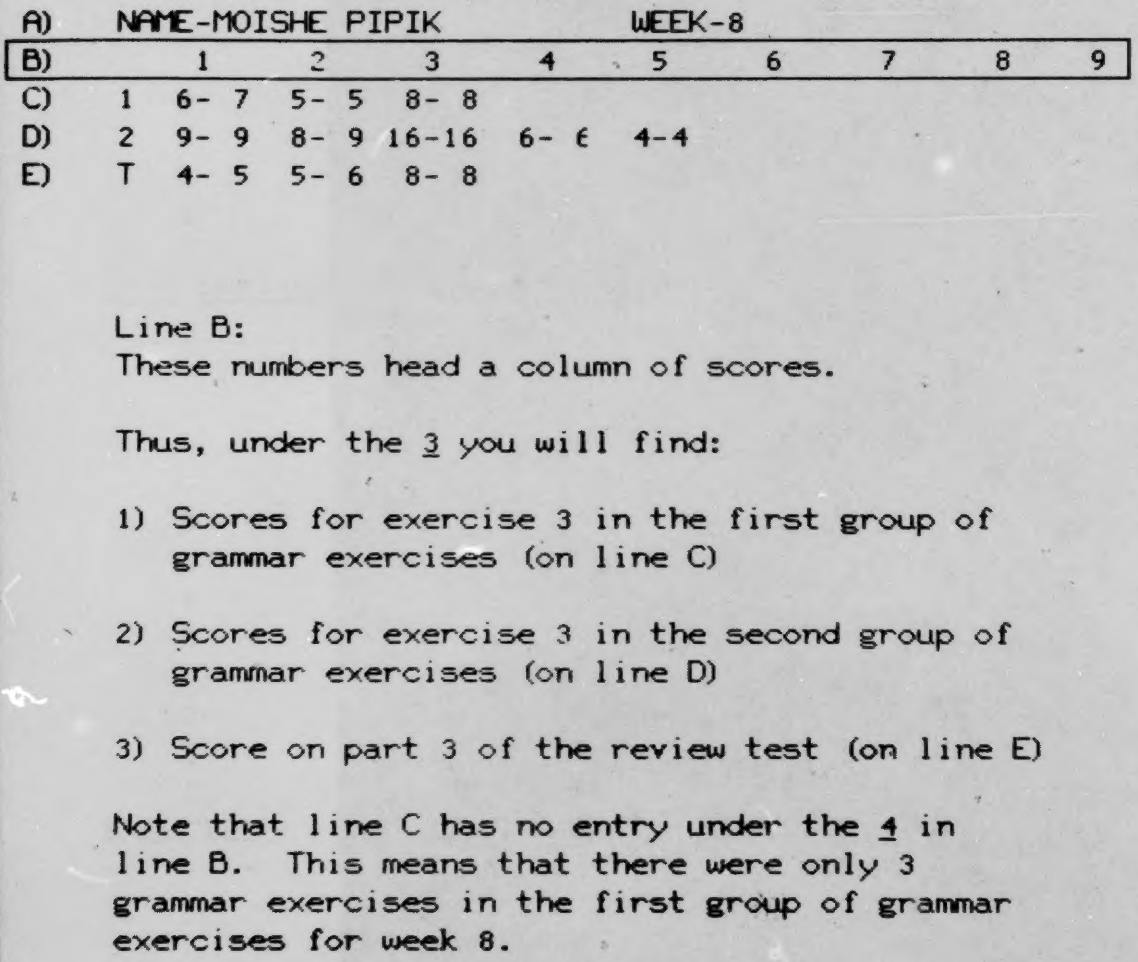
ACROSS:
a5: grandfather
b6: a friend (m)
c6: a daughter
c3: slowly
d6: first
e2: opposite of כאן
f5: sixth (m)
g6: her
g2: what

Start at: ⪢
-LAB- to check, -DATA- when done.

Figure 5

```
STUDENT: yael

SUMMARY FOR WEEK: 10


                        PART 1              PART 2
EXERCISE              SCORE  MAX         SCORE  MAX
    1                  10     10           6      6
    2                   8      8           9     10
    3                   7      8          10     10
    4                                      4      4
    5                                     13     13
    6
    7
    8
    9



TEST SECTION           SCORE      MAX
    1                    9         13
    2                   14         14
    3                    6          6
    4
    5
```

Figure 6



```
A)    NAME-MOISHE PIPIK          WEEK-8
B)        1     2     3     4     5     6     7     8     9
C)    1  6- 7  5- 5  8- 8
D)    2  9- 9  8- 9 16-16  6- 6  4-4
E)    T  4- 5  5- 6  8- 8
```

Line B:
These numbers head a column of scores.

Thus, under the 3 you will find:

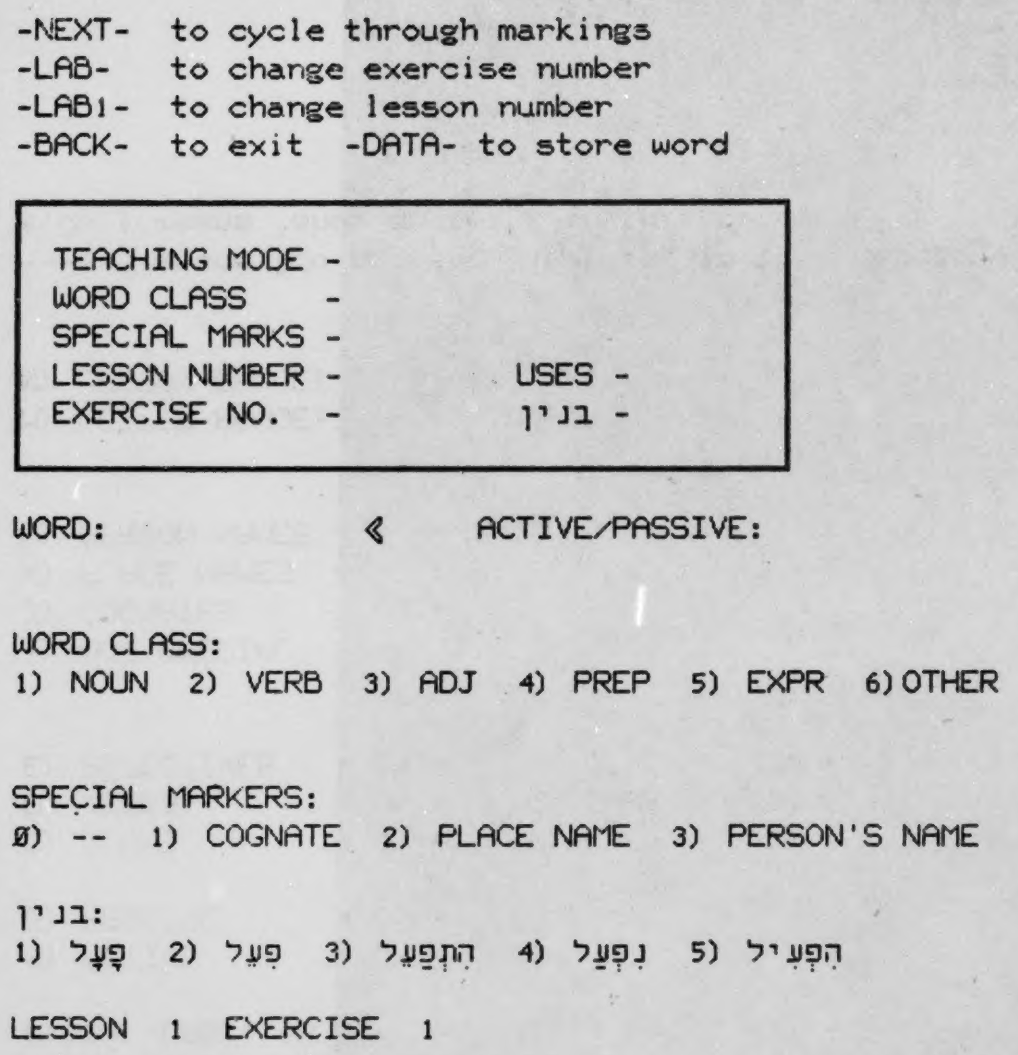
1) Scores for exercise 3 in the first group of grammar exercises (on line C)

2) Scores for exercise 3 in the second group of grammar exercises (on line D)

3) Score on part 3 of the review test (on line E)

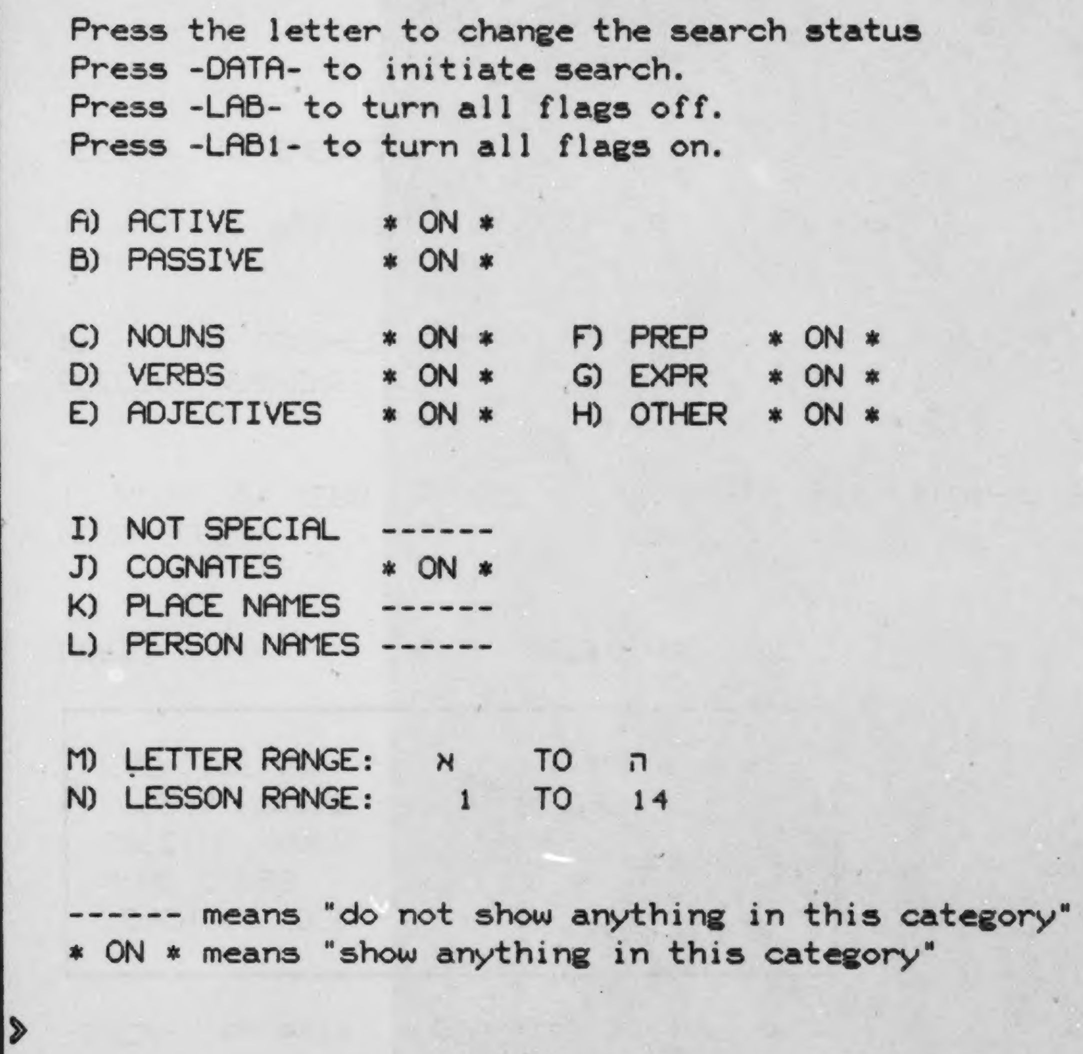
Note that line C has no entry under the 4 in line B. This means that there were only 3 grammar exercises in the first group of grammar exercises for week 8.

Figure 7



```
-NEXT-   to cycle through markings
-LAB-    to change exercise number
-LAB1-   to change lesson number
-BACK-   to exit  -DATA- to store word

 TEACHING MODE -
 WORD CLASS    -
 SPECIAL MARKS -
 LESSON NUMBER -          USES -
 EXERCISE NO.  -          בנין -

WORD:                 «     ACTIVE/PASSIVE:

WORD CLASS:
1) NOUN  2) VERB  3) ADJ  4) PREP  5) EXPR  6) OTHER

SPECIAL MARKERS:
0) --  1) COGNATE  2) PLACE NAME  3) PERSON'S NAME

בנין:
1) פָּעַל  2) פִּעֵל  3) הִתְפַּעֵל  4) נִפְעַל  5) הִפְעִיל

LESSON  1  EXERCISE  1
```

Figure 8



```
Press the letter to change the search status
Press -DATA- to initiate search.
Press -LAB- to turn all flags off.
Press -LAB1- to turn all flags on.

A) ACTIVE          * ON *
B) PASSIVE         * ON *

C) NOUNS           * ON *    F) PREP    * ON *
D) VERBS           * ON *    G) EXPR    * ON *
E) ADJECTIVES      * ON *    H) OTHER   * ON *


I) NOT SPECIAL   ------
J) COGNATES        * ON *
K) PLACE NAMES   ------
L) PERSON NAMES  ------


M) LETTER RANGE:    א     TO    ה
N) LESSON RANGE:     1    TO   14


------ means "do not show anything in this category"
* ON * means "show anything in this category"

>
```

Figure 9

```
 1.        אגורה    ACTIVE   NOUN   COGNATE      10- 5
 2.         אוטו    ACTIVE   NOUN   COGNATE       4-15
 3.      אוטובוס    ACTIVE   NOUN   COGNATE       7-19
 4.        אולפן    ACTIVE   NOUN   COGNATE      11-26
 5.   אוניברסיטה    ACTIVE   NOUN   COGNATE       2-11
 6.        אידיש    ACTIVE   NOUN   COGNATE      11-24
 7.      אמבולנס    ACTIVE   NOUN   COGNATE      13-34
 8.  אנציפלופדי    ACTIVE   NOUN   COGNATE      10-12
 9.     אקורדיון    ACTIVE   NOUN   COGNATE      12-10
10.          בוס    ACTIVE   NOUN   COGNATE       5-35
11.     ביולוגיה    ACTIVE   NOUN   COGNATE       3-33
12.       ביקיני    ACTIVE   NOUN   COGNATE       9-28
13.         בירה    ACTIVE   NOUN   COGNATE      10- 6
14.         גרג׳    ACTIVE   NOUN   COGNATE       9- 7
15.         דולר    ACTIVE   NOUN   COGNATE       8-32
16.       דוקטור    ACTIVE   NOUN   COGNATE       9- 6
17.          הלו    ACTIVE   EXP    COGNATE       8-29
18.      המבורגר    ACTIVE   NOUN   COGNATE       1- 4
19.      הסטוריה    ACTIVE   NOUN   COGNATE       2-11



                ** END OF WORDS **
```

Figure 10

OPTIMIZING COURSEWRITER PARAMETERS TO REDUCE RESPONSE TIME AT REMOTE SITES

R. Lindwall
Programmer/Analyst
School District of Kansas City, Missouri
1211 McGee, Room 610
Kansas City, Missouri 64106
(816) 221-7565 extension 221

In 1974, the School District of Kansas City, Missouri, expanded its Computer Assisted Instruction (CAI) program to a second junior high school. Coursewriter III was installed as the controlling program in the central office computer (IBM 370/135) using parameter values suggested in installation manuals. Remote dedicated telephone lines with 2400 bit per second (BPS) modems and 3271 control units send data to the two sites with seven and fifteen 3277 (480 character) terminals respectively. An Integrated Communications Adapter (ICA) was used to connect each line to the computer.

With this configuration, the response time at the schools was adequate and the achievement gains by the students were impressive.

The School District expanded the CAI system in October, 1975, to two elementary schools with eight terminals each, on a multidrop 2400 BPS line. The internal machine response time increased to over three seconds. The average response time at these new schools was 10.98 seconds, and only 54.83% of the response times were less than ten seconds. This was excessive but still tolerable. There were, however, occasional periods of severely degraded response times, in this case in excess of 60 seconds. This was intolerable and attempts were now made to determine the causes and to effect changes.

During the next six months responses were timed, parameters changed, analysis programs written, Coursewriter program coding changed, statistics compiled, dumps analyzed, and hardware changed. As a direct result of these modifications, guidelines have been developed for analyzing and optimizing the performance of Coursewriter on a system. These guidelines consist of suggestions for installing a new system and for improvements which can be made to existing installations.

Section 1. Drop the Multidrop

The first timings at the schools showed clearly that whenever students were using terminals at both of the new schools simultaneously, the response times at those two sites increased dramatically regardless of other system activity, and regardless of response times at the two original sites. (See Figures 1, 3, and 4.) It was apparent that this was due to the fact that both of the new schools were served by one multidrop telephone line, whereas the two junior high schools were served by two separate telephone lines.

Several telephone calls were made to IBM CAI experts about this phenomenon and they replied, after some study, that Coursewriter was indeed inefficient when dealing with multidrop lines, but to alleviate this problem would require a major rewrite of the Coursewriter controlling programs, and they were unwilling to do this. They suggested a separation of the multidrop telephone lines - a simple, relatively inexpensive hardware addition.

In order to separate the telephone lines, it would be necessary to install a fourth integrated communication adaptor (ICA) at an approximate one time cost of $2,400.00 (because the School District had purchased its computer, IBM would not lease any internal component for it). Other needs were an additional modem at $106.00 per month, and the fourth telephone line at $39.00 per month.

At this time, IBM supplied programs to monitor several of the machine functions. These programs analyzed disk activity, file placement, interpartition contention, and channel usage. These programs also analyzed ICA usage and convinced the manufacturer's highly experienced technical representatives that ICA contention was itself a major factor in the overall slow response times. The School District staff was persuaded that adding a fourth ICA would only aggravate the problems, and IBM representatives recommended that the District lease a 3704 line controller at $1,084.00 per month to handle the 39 Coursewriter terminals on four telephone lines.

After a meeting to analyze the information available at that time, an order was placed for a line controller, a modem, and a fourth telephone line.

Several months later, when the new hardware arrived, the installing field engineer took one look at the line controller and asked why in the world that machine had been ordered, since it certainly would not alleviate the problems at hand.

Two separate timing sessions were conducted as the new equipment was installed in stages. First, the line controller was installed. It made no difference that could be detected in stopwatch response times, or internal machine response times. (See Figure 4, timings of 4/9/76.) The lease was canceled for this line controller.

While this line controller was in operation, a fourth telephone line could be added without purchasing an ICA. The multidrop lines were separated and there were then individual dedicated telephone lines to each of four schools. The next timing session showed that the response times at the two elementary schools decreased sharply, and for the first time the response times at each site were directly dependent on the number of student terminals in use at a given site at any particular time. (See Figure 4, timings of 5/3/76.)

A comparison of Figures 1 and 2 will aid in visualizing the impact this change had on response times at these two elementary schools.

Section 2. Probe the Depth

In March of 1976, it was determined that the DEPTH parameter of the DEFBLKS module did not function properly with 480 character screens and 2K LDA areas. Without DEPTH specified, the system would display one line for each write (12 writes to fill the screen). With DEPTH specified, the system would display four lines for each write (3 writes to fill the screen). Eventually, it was made to write a full screen by generating 4K LDA areas (timings 9/23/76). This allowed, for the first time, stopwatch timings of less than two seconds (see Figure 5). It also increased system size, paging rate, and LDA read/write time so much that the average internal machine response time was increased by about 70 percent (see Figure 6).

A resolution has finally been achieved for eliminating these problems (APAR WP 6102) and has considerably decreased the average stopwatch response times (timings 11/3/76) and returned the internal machine response times to the level they were prior to generating the 4K LDA areas.

Section 3. Squeeze the Subroutines

A program was written to analyze the frequency of usage of each user written function subroutine. Subsequently, many of these functions were concatenated to optimize core usage and diminish calling frequency. The number of active 2K functions was reduced from fourteen to eight, which was expected to have some noticeable effect on response times. It did, but only a very slight effect. It was subsequently determined that the functions at this school district installation comprise only about five percent of the subroutine calls.

The next step was to redistribute and combine Coursewriter system subroutines. As each new combination was implemented, a definite decrease in the internal machine response time was noticed. The thirteen subroutines identified in the system documentation as being used by student processing routines have been combined in such a manner that they now comprise only eight subroutines. These operations alone have reduced the needed subroutine fetches from disk by 70 percent, without changing the number of subroutine areas in core. Four more subroutines have subsequently been found to be used in signing students on and off the system. Three of these four have been combined in such a manner that only one is now used by students, leaving the other two inactive. The fourth is as yet unchanged.

Coursewriter now has ten subroutines used by students, any additional subroutine areas can be used by user functions. Combining these subroutines and functions has reduced the number of subroutine fetches from disk by 90 percent, improved the usage of memory, and has been substantially responsible for the greatly reduced internal machine response times.

Section 4. Peripherals and Page Faults

Many thousands of instructions can be executed in the same length of time any Input/Output request can be satisfied. Because of this, it is beneficial to optimize the usage of and number of I/O requests and I/O areas.

Page faults cause the partition to wait until they are satisfied. Coursewriter will attempt to continue processing other phases of its programs while an I/O request is being satisfied in another section. The LDA and Subroutine files are direct access files and the record needed can be accessed with no necessity for multiple reads. These two files may be good candidates for fixed head disk files because of their efficiency and small size. This might allow specification of fewer LDAS and SUBS while reducing the Coursewriter working set and therefore the paging rate.

Paging activity is a justifiable concern with Coursewriter because it has a very large working set (about 90 percent of linkage size) as determined independently at two installations in Kansas City.

Section 5. Module Monitoring

DEFBLKS module.

PAGING and HARDCPY are very costly options. They cause the generation of 4K LDA's and require DEFCAI5, both of which increase the Coursewriter working set considerably, and increase paging. Each of these options requires a separate disk file with a considerable number of I/O requests. Generating 4K LDA's, without using either paging or hardcopy, increased the internal machine response time by approximately 70 percent. The usefulness of these options should be carefully evaluated on any system.

DEPTH will significantly reduce time required to write a full screen at no apparent cost.

BUFS are necessary for teleprocessing I/O. A formatted dump of Coursewriter can be used to determine number of write areas actually necessary (this may vary greatly from day to day). Each BUFS specified takes 320 bytes of core (128 bytes for TP buffer plus 1½ times that for output buffers).

SUBRS are important to optimize. Each two additional subroutine areas will decrease subroutine reads approximately 50 percent. Authoring and supervisor requests use subroutines not used by students. Any installation which has considerable usage by authors or supervisors while many students are on line will need more subroutine areas, and should consider concatenating some of these subroutines.

LDAS are difficult to optimize, but also very important, and relate to the number of terminals in use simultaneously. As LDAS are added or subtracted, it is important to watch the internal machine response time, percentage over five seconds, as well as the relationships between ROLI, RCLM, ROLO, and RLSE.

TIMER is in the OPTN macro, and has no noticeable effect on the system at this school district.

DEFDBUG module.

Counters can be a very valuable tool for monitoring and tuning Coursewriter. They also add 1700 bytes to the system size.

Formatted dumps can be a valuable aid in debugging, monitoring and tuning Coursewriter. They also add 2400 bytes to the system size.

Either of these parameters requires SYSLYST to be assigned, with all the associated output requests.

DEFAMOD module.

A system which does not use a printer does not need a PRMOD.

A system which does not use OLTEP can specify BSCTEST = NO.

A system which does not use teletype, 2740, etc., can specify SS = NO.

A system which infrequently uses tape might place MTMOD at the end of the module.

Section 6. Other Considerations for Coursewriter

User written functions may be loaded into the subroutine file to put the most frequently used functions at the beginning of the file to minimize sequential table searching for function name. This may be done by loading the most important modules individually, then loading all others together.

WTAP will normally indicate the number of writes to the Student Response File. An evaluation of the use of this information might suggest a selective use of the recording option when registering or placing them under author control within the course material by using functions TYREC and RECORD from the Ohio State University.

The storage requirements of several system modules should be examined because several of them can probably be eliminated from any system (DEFWORK for example).

The calculated time required for transmission of 480 characters over a 2400 BAUD telephone line (300 characters per second) is 1.6 seconds. Faster modems (i.e. 4800 BAUD) would decrease this minimum to 0.8 seconds.

Section 7. Considerations for other Partitions

Dynamic allocation of core for sorts and compiles make them good candidates for the use of a SIZE parameter on the execute card.

Partition priority may be a factor at some installations.

File placement can in some cases aid in avoiding disk contention between Coursewriter and other partitions. Some batch jobs might best be run at times of light CAI activity, or at night.

In summary, the three major changes consisted of:

A. Separating the telephone line from one multidrop line to two individual telephone lines. This greatly affected the response times at the elementary schools (Holmes and Franklin), without affecting the junior high schools in any way.

B. Combining and optimizing the functions and subroutines reduced the number of subroutine reads by 90 percent, reduced the minimum stopwatch response time from three to two seconds, and helped the entire system.

C. Repairing the DEPTH parameter has significantly reduced the minimum stopwatch response time to less than two seconds, and helped the entire system.

A goal of having 90 percent of all stopwatch responses to be less than ten seconds has been suggested as a criterion for reasonable response time. When this project was initiated in November of 1965, only about seventy percent of all responses at Bingham and Central were less than ten seconds, and the average time was 8.81 seconds. At the present time, about 99 percent of all responses at these two schools are less than ten seconds, 90 percent are now less than 4.40 seconds, and the average stopwatch response time at Bingham and Central is currently 2.79 seconds (which is less than the minimum response time was a year ago).

Conclusion

The viability and longevity of any system depends on its uniqueness, accomplishments, and efficiency. The life of Coursewriter at the School District of Kansas City, Missouri, has been extended by optimizing controlling program code in such a manner as to significantly reduce response times on the student terminals, and reduce the processing load on the computer to allow greater utility of our multiprocessing computer. This has also given valuable experience in anticipating further expansion of a Computer Assisted Instruction system.

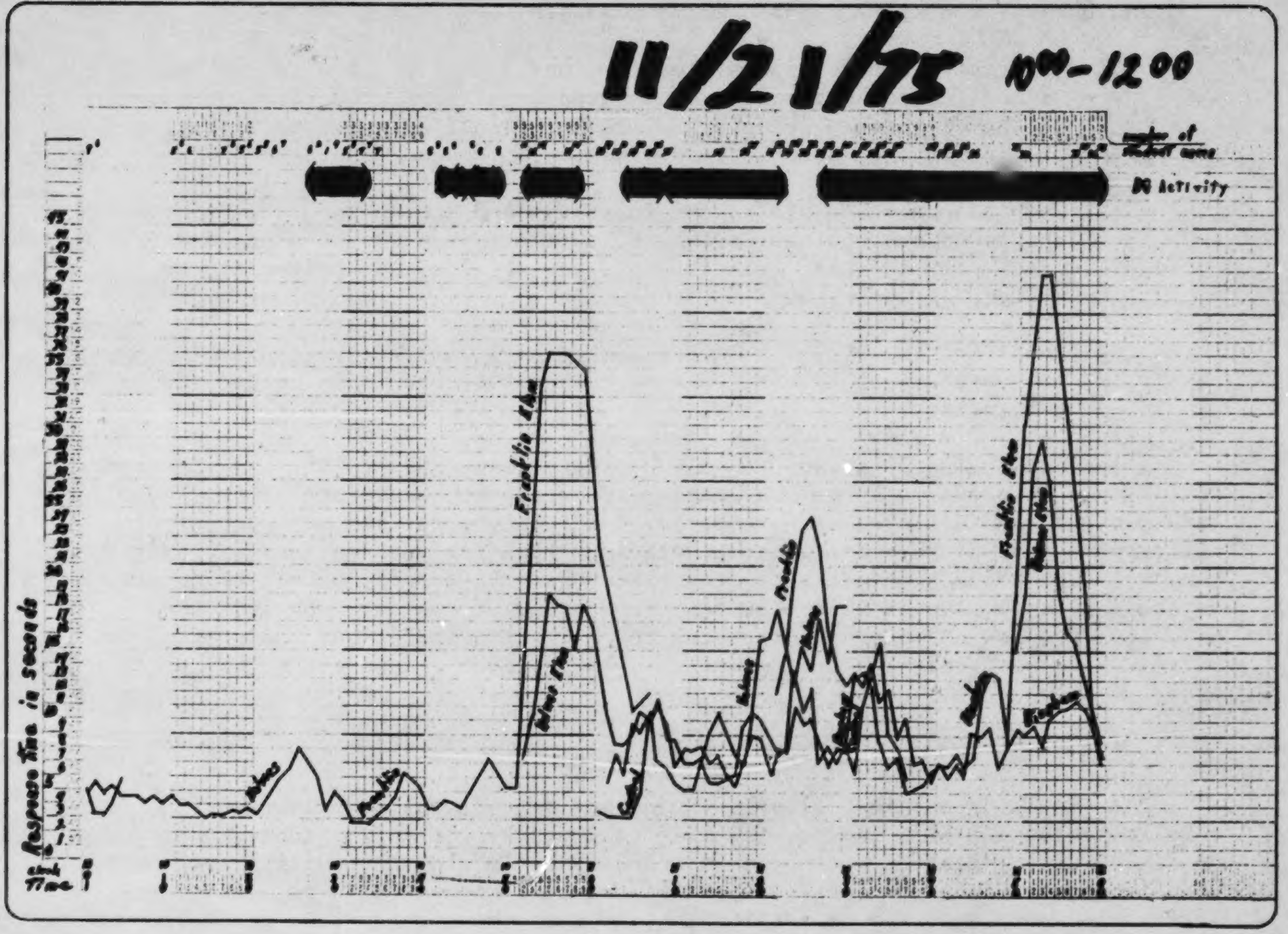


Figure 1

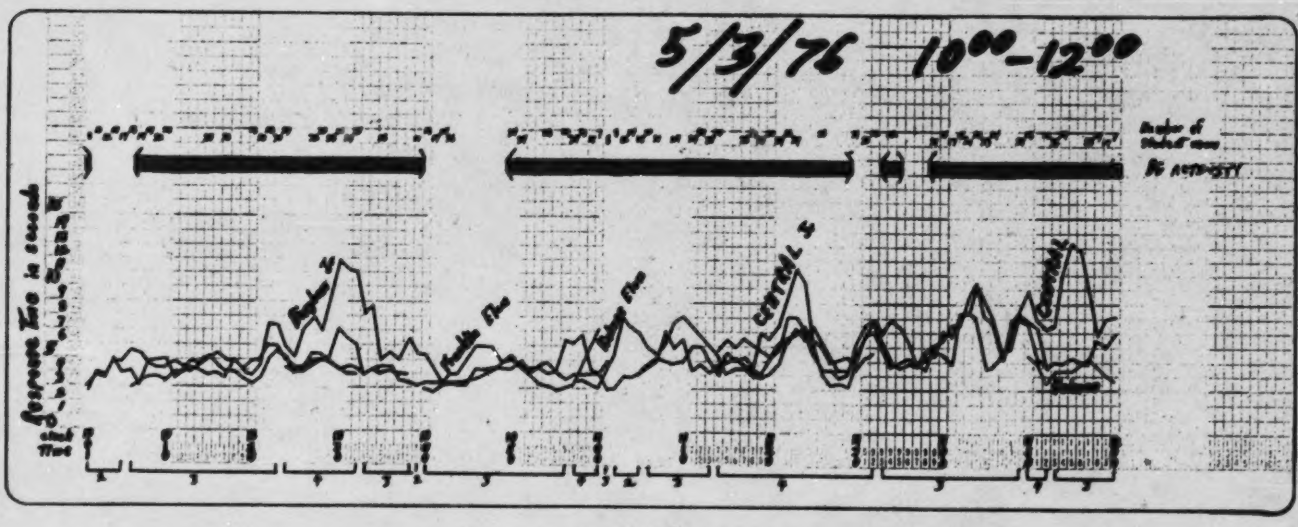


Figure 2

November, 1975

	Holmes and Franklin Multidrop	Bingham and Central Non-Multidrop
Number of Terminals	16	22
Average Response Time	10.98 sec	8.81 sec
Percentage Under Ten Seconds	54.83 percent	69.20 percent
Ninety Percent of Responses Less Than	23.00 sec	16.30 sec

Figure 3. Full day averages for multidrop and non-multidrop lines.

		Multidrop				Non-Multidrop			
		Holmes		Franklin		Central		Bingham	
DATE	DURATION	% < 10	avg sec	% < 10	avg sec	% < 10	avg sec	% < 10	avg sec
11/14/75	29 min	27.59	12.59	24.14	15.00	86.21	6.77	79.31	7.48
11/14/75	48 min	inactive		inactive		89.58	5.42	83.33	5.73
11/14/75	11 min	9.09	18.45	25.00	25.75	inactive		63.64	9.91
11/21/75	23 min	91.30	5.91	inactive		inactive		95.65	5.09
4/9/76*	19 min	26.32	12.66	23.53	20.47	80.00	7.73	68.42	11.18
5/3/76**	19 min	100.00	5.21	94.74	6.34	84.21	7.92	89.47	5.68
11/14/75	4 hrs	56.35	10.20	53.28	11.80	81.25	6.76	76.84	7.24
5/3/76**	4 hrs	98.47	4.50	92.11	5.55	89.15	5,83	89.44	5.80

Figure 4. Response times at Holmes and Franklin are greatly degraded when students are on at both of those sites at once.

*Timings made to determine effectiveness of Line Controller.
**Timings made to determine effectiveness of separate telephone lines.

Date of Timings	Minimum Stopwatch Response Time
11/14/75	3.0
11/21/75	3.0
12/18/75	3.0
2/26/76	3.0
3/22/76	2.5
4/9/76	2.5
5/3/76	2.5
9/21/76	2.0
9/23/76	1.5
11/3/76	1.5

Figure 5. This minimum response time is indicative of the improvement in the efficiency of the controlling program, and would be representative of the absolute fastest possible response with only one terminal operational, with no activity in background. The frequency of appearance of this minimum time is related to the number of terminals in use and background activity.

Time Period	Average Internal Machine Response Time	Number of Terminals Active
10/14 - 11/13/75	2.54	38
11/14 - 12/18/75	2.38	38
1/6 - 2/11/76	1.80	38
2/13 - 3/5/76	1.53	38
3/8 - 3/29/76	1.42	38
3/30 - 4/21/76	1.28	38
4/22 - 5/11/76	1.14	38
*Summer School	.70	16
(4K LDA) 9/22 - 10/25/76	1.51	22
*10/27 - 11/24/76	.90	22

Figure 6. This table indicates a gradual increase in efficiency of the controlling programs as a result of the cumulative effect of many small changes.

*Response time reduction is largely due to fewer terminals in use.

Response Time in Seconds	11/21/75	5/3/76	9/21/76	9/23/76	11/3/76
0 - 0.9	0	0	0	0	0
1 - 1.9	0	0	0	4.31	28.00
2 - 2.9	0	3.82	29.88	37.64	70.40
3 - 3.9	12.11	28.82	58.62	62.35	85.60
4 - 4.9	21.79	50.25	70.11	77.64	*92.80
5 - 5.9	35.64	66.07	77.58	87.84	95.60
6 - 6.9	47.05	76.02	82.18	*92.54	97.20
7 - 7.9	54.32	80.86	85.05	95.29	98.40
8 - 8.9	64.70	85.20	89.65	96.47	99.60
9 - 9.9	69.20	89.28	*91.95	97.64	99.60
10 +	100.00	100.00	100.00	100.00	100.00

Figure 7. This table shows the percentages of stopwatch timings which were within a given range or lower. Representative samples are shown here for Central Junior and Bingham Junior combined (the multidrop line is not included).

*90% of the responses were lower than this.

Number of Active Schools	Approximate Number of Terminals	5/3/76 Average Response Time (Sec)
1	10	4.41
2	19	4.91
3	29	5.35
4	38	5.71

Figure 8. This table indicates there is a definite and significant positive correlation between number of active terminals and increased response time. We do not have enough information to determine whether this is a linear or logarithmic relationship.

APPENDIX A. Activity Counter Totals.

If the CNTRS = YES option was specified for the DEFDBUG module, activity counters for several categories can be printed on SYSLST by requesting the COUNTERS, EOJ, or DUMP option on the system console. These listings may be retained for the valuable information they contain, or the most pertinent data may be transferred to a simple chart to facilitate comparisons on a day to day basis to monitor the effects of system changes. The following information has been found to be most useful:

RTEL
WTEL
RDSK
WDSK
SCAL
SFET

WTAP
RLSE
RCLM
ROLO
ROLI

CMPL
FN
SR/W
CM/S
CM/A

0-3%
5+ %
system average
total counts

Other information worth recording is: date, system available time, and maximum number of users of each type (from the line activity messages also recorded on syslst).

In some cases, the relationship between two counters can be more easily compared than can the counters themselves.

$$\frac{\text{SFET}}{\text{SCAL}} \qquad \frac{\text{RLSE}}{\text{ROLI}} \qquad \frac{\text{RCLM}}{\text{ROLO}} \qquad \frac{\text{ROLI}}{\text{RCLM}} \qquad \frac{\text{ROLI}}{\text{SFET}} \qquad \frac{\text{FN}}{\text{SFET}}$$

APPENDIX B. Dump Information

In the FORMDMP = YES option was specified for the DEFDBUG module, the DUMP option on the system console can provide information in a ready-to-use form on SYSLST (or a link map can be used with a system dump).

The actual subroutines in use, and over a period of time their frequency of usage can be determined by recording the contents of the subroutine areas SRBLK1 through SRBLKNN (SRBLKS on link map).

```
                                     F  N
SRBLK11   08E010   47 F0 20 10   C6 D5 |6B 73|
          08E030   E8 D9 C9 C7   C8 E3 40 C9        user written function
          08E050   C5 40 D5 E4   D4 C2 C5 D9

                                     S  R
SRBLK10   08D810   47 F0 20 16   E2 D9 |00 08|
          08D830   40 04 47 80   21 DE 95 3C        system supplied subroutine
          08D850   40 04 47 80   23 9A 91 08
```

The number of TP buffers actually used can be determined by counting the number of TPB areas occupied by code (TPBUFS on link map). The Coursewriter link-edit must specify "action clear" to simplify the distinction between used and unused areas.

```
TPB10     086E90   C1 01 01 01   01 01 3D 3D
          086EB0   3D 3D 3D 3D   3D 3D 3D 3D
          086ED0   3D 3D 3D 3D   3D 3D 3D 3D        used TP buffer
          086EF0   3D 3D 3D 3D   3D 3D 3D 3D

TPB11     086F70   00 00 00 00   00 00 00 00        unused TP buffer
```

APPENDIX C. Stopwatch Timings.

Part 1. Procedure for Recording Response Times

A. Sign on as supervisor, press enter key.
B. Type 'LINES' and press enter key on console.
C. Activate stopwatch at same time as you press enter.
D. When 'TYPE COMMAND' (last line) appears on terminal, stop stopwatch.
E. Read elapsed number of seconds on inner (1-60) dial of stopwatch.
F. Record time last passed to accuracy desired (1/2 or 1/10 sec).
G. Read time of day from top center of screen.
H. Record elapsed number of seconds in response time column adjacent to appropriate time of day on chart.
I. Press reset button on watch.
J. Press reset and clear keys on console to clear screen.
K. Go to 'B' above. Repeat once each minute. Refer to watch or wall clock to maintain uniform one minute intervals.

NOTE 1: Each time you press "enter" key, check that input inhibited indicator light appears on screen, if not, reset stopwatch and press enter again.

NOTE 2: 'LINES' was selected as a standardized command for use because it fills the 480 character screen, usually requires a subroutine read, and is readily available to every Coursewriter user.

Part 2. To assist in analyzing the information recorded, the information on the console log, syslst, and timing sheets can be compiled onto response charts (see Figure 1).

Part 3. This is an example of a timing sheet which can be used to record timings and calculate averages.

LINES DATE__________ LOCATION__________

Time of Day	Response Time

This chart can be made to accommodate 60 response times.

Total responses > 10 sec -----
this page < 10 sec -----

Total time
this page ----- sec

Total responses > 10 sec -----
all pages < 10 sec -----
Total responses -----

Total time all pages ----- = ----- average response time
Total responses -----

Total responses < 10 sec ----- = -----% < 10 sec
Total responses -----

A COMMUNITY OF INDIVIDUALS:
COOPERATION AND INDIVIDUALIZATION
IN COMPUTER SCIENCE EDUCATION

Kenneth Modesitt
Indiana University - Purdue University

Introduction

In this paper, the concepts of community and the individual are compared and contrasted. The realization of these concepts has been implemented in the classroom by maintaining a delicate balance between cooperative efforts and individualized instruction. Preserving this balance is advocated as a preferable alternative to either the traditional lecture mode or, more recently, the primarily individualized mode of instruction. Considerable experience has been gained with the individualized mode, utilizing both personalized systems of instruction (PSI) and computer-based education (CBE) over a period of several years. The explicit introduction of cooperation into the computer science classroom is of more recent vintage. It comes in response to a belief that cooperative efforts are most likely to solve the difficult problems of today, and to fears of interpersonal isolation which may result from exclusive or primary reliance on individualization.

Definitions [17]

Community:	a social group or class having common interests
Common:	belonging equally to two or more; shared by all alike; joint
Cooperate:	work together toward a common end or purpose
Cooperation:	an association of persons for mutual benefit
Individual:	of or relating to a single human being; by or for one person
Individualize:	to modify or suit a particular entity

Why is a Community of Individuals Desirable?

Whenever I meet a class for the first time, one of the first questions I ask them is: "Why?" "Why are you in this course?" "Why are computers becoming such an integral part of our lives?" "Why is understanding their uses and potential so important?" To me, asking "Why" is an essential first step in looking at a new idea. No particular solution is guaranteed to result. Rather, oftentimes, an increased awe and reverance for the phenomenon results. Nevertheless, when the phrase "community of individuals" first began to have some significance for me as an instructor, I needed to ask, and perhaps answer, the "Why" question.

I believe that a community of individuals, working together in cooperation toward mutual goals represents the most hopeful and realistic structure whereby the majority of the important problems confronting us today can be solved. The difficulties and enigmas which surround us today are legion and increasingly severe. For cogent and striking presentations of the present and possible futures, recent issues in The Portable Stanford Series are outstanding [6, 18, 20]. Of particular import is Harman's An Incomplete Guide to the Future. Wars, unemployment, famine, inflation, civil rights, crime, welfare, corruption -- the list appears endless.

Few of these, if any, are solvable by one person. What are the alternatives: to continue to accept such problems as "natural" or treat them as "inconveniences"? I, at least am not willing so to do. Rather, it is my firm belief that a dedicated community of strong individuals offers a viable and desirable alternative.

Alternative Models in Education

The belief in a dedicated community of strong individuals is manifested within the framework of introductory computer science sources at Indiana University-Purdue University at Fort Wayne. I strongly believe in the impact that models have on our later lives. Hence, I try to include in classroom situations those values which have positive significance for me, i.e., community of and cooperation among mature persons. It is this model I wish students to consider seriously now and in later life.

Students will definitely have other models from which to choose. One of the most common and powerful ones is based on competition, and is usually found in a traditional lecture-oriented course. In earlier papers, my thoughts were shared about where and when competition may be appropriate, e.g., sports, and where it is not appropriate, e.g., education [13, 14]. In a recent address, Howard Casmey, Minnesota commissioner of education, noted that education today is "predicated on failure" [1]. However, a grade of A or B in a course must indicate that a student has demonstrated mastery of the subject. It must not indicate that she/he is in the top 10% who demonstrate mastery in a predefined, usually short, time frame, e.g., an hour exam, a two hour final, a one semester course, or a four year degree, etc.

So we are led naturally to another, and rapidly growing, model in education: that of individualization. To again quote Casmey,

> "To me 'individualized instruction' connotes 'mastery learning.' It's neither a concept nor a process in and of itself, but rather a concept that requires a process. It is the great humanizing factor in education, because failure is no longer possible. All students now succeed; some at a faster or slower rate than others, but all learn and all succeed."

> "If one of our children has more difficulty in learning to tie his shoe, and does not accomplish this task until he is six years old, we do not label the child as a failure. We know that, barring any specific physical problems, he will be able to accomplish this task as his manipulative skills develop."
>
> "This is what the educational process should be and we've known this for 50 years."

(Underlining mine)

Casmey goes on to espouse the use of computer-based education. Personalized systems of instruction are also vitally concerned with individualized instruction. This model is leading to persons increasingly able to become adults, i.e., persons who assume responsibility for their own actions. It is a model I view with great respect and admiration. I have used it for the last four years in several courses. It continues as a vital and integral part today. However, and here I quote the frustrated and honestly concerned housewife who, when asked about her housework, replied, "It's good, but it's just not enough!" As I recall the story, she also went on to say she needed someone to share in those responsibilities.

My feelings about individualized instruction are similar: it's good; I like it; I encourage my students to use it; but it is simply not enough! The process does promote a more adult, independent attitude. Students do gain an acceptable and accurate self-image: "I can learn this subject pretty much on my own." To many people, young and old alike, this is a dramatic change from "I'm dumb. I never could do math (or English or physics, etc.)." In fact, it was the latter type statement that first prompted me to consider seriously a major shift in my courses from lecture to individualized.

However, it has been several years since that time, and I've come to realize that strong individuals, by themselves, do not constitute a complete picture. You and I really do live in a community, several of them in fact. I do not wish to live a life apart from others. I wish to participate, as a strong person, in mutual efforts with the other strong people to solve significant problems and to share with them at increasingly open and intense levels. The latter quality is not a "teachable" one in my opinion, but rather flows naturally from significant mutual content-based concerns. Therefore, it is this model of a dedicated community of responsible individuals that I value highly, and consequently emphasize in many of my courses.

Cooperative Course Efforts

The course in which major efforts have been expanded to incorporate cooperation is CS 220, Introduction to Algorithmic Porcesses. Section enrollment is usually between 20 and 25, and consists primarily of science majors ranging in age from 18 to 30. It represents the first direct exposure to computers for all those with some experience in high school. A course which several take after CS 220 is CS 461, Algorithmic Languages, which relies extensively on individualized instruction.

The primary cooperative efforts in CS 220 include the following: study partners for PSI units, data base design by small groups, mutual design and use of interactive programs, use of cooperative exercises, and parties.

1. Study partners for PSI units

As most of you know, a personalized system of instruction, also known as the Keller plan, has become one of the most noteworthy and exciting instructional methods in recent years. Earlier referenced papers of mine discuss it at some length, as well as in [15, 16]. It is appearing in more and more university settings, both as documented from within and without. From within, the Center for Personalized Instruction at Georgetown University holds annual conferences [21]. They also publish a newsletter [19] and a journal [11]. All these publications relate the increased use and very positive results in final exam scores, retention, transfer, attitudes, facilitation, and self-image. From without, recent issues of Change magazine contain several articles on PSI as one of the most promising innovative educational concepts [4, 12].

An integral part of PSI is breaking material down into digestable pieces or units. For CS 220, there are 15 such units (two of them review ones), plus a final. Within the normal PSI framework, each student works through the objectives, suggested procedure, and unit test by herself/himself. Within the cooperative PSI framework, each student is assigned a partner on the first day of the course. The partner is usually of the opposite sex, roughly the same year in school, but not necessarily in a related discipline. During the next couple of class days, the partners introduce each other to the class informally.

The PSI partners are encouraged to work together on the objectives and suggested procedure of each unit. The evaluation, of course, is administered separately. Since most of the units involve designing and implementing computer programs, a variant of "egoless" programming is suggested [2].

The partnership is particularly beneficial in the first several weeks. The intricacies of working with strange machines and procedures are not as intimidating to two people as to one. Therefore, when the units become more concerned with concepts instead of interactive terminals, batch jobs, remote job entry, keypunches, etc., partnerships may deepen, dissolve, or reform. By this time, the partners are often working at different rates. So, they may feel free to work with others, or in some cases, by themselves. In any case, all the students now have first-hand experience in cooperating with another for a common goal.

As those of you familiar with PSI know, the peer tutors or proctors are to perform many of the rolls just mentioned. The three peer tutors for CS 220 are excellent, in my opinion. The students view them as "willing-to-help" experts, and have no hesitation about asking them questions. However, the tutor is not a true partner - she/he has been through the course earlier, whereas the student is still working on it. Hence, I believe both the tutor and partner have valuable roles to play in the learning process.

2. Automated personal data base design by small groups

After nearly one-half of the semester has passed, most students have a working knowledge of the primary programming language concepts. It then becomes feasible to introduce them to a design project. Since such projects outside the academic world are usually cooperative in nature, this is a good opportunity for small groups of 3-4 to work together. I wanted the design to be for a realistic area and one of some importance. Automated (computerized) personal data banks immediately come to mind. A group in the College of Education at the University of Illinois under the direction of Bruce Hicks has also investigated such issues [9]. In CS 220, however, the students are not only users of such data banks, they will also be designers thereof.

The sample list of such data banks included in the appendix is almost solely student-generated. Here is an application in which interest is quite high. Outside speakers come in to discuss the issues of privacy and society's need for organization. Students have the necessary skills to build such a data base. The task lends itself readily to sharing ideas and feelings, especially about one's right to privacy. The complete assignment is given in the appendix.

3. Mutual design and use of interactive programs

After some introductory simple programs which all students are to complete, students are given a choice. They may either work on problems assigned in the unit, or they may choose one of their own, so long as it is of comparable complexity. Several people have opted for the latter. Here they can use the computer directly for a problem in their discipline, or they can design and implement a game, simulation, drill and practice, etc.

In any case, whether the students work on a preassigned problem or one of their own, they are encouraged to seek out an "intelligent but naive" classmate to be a typical user. This user, who is often their partner, takes the program. She/he writes comments relating to directions, accuracy, style, etc., and then signs and dates the listing. She/he is not expected to examine the coding itself. If necessary, the author then modifies the program before turning it and the signed listing in as part of the unit test.

Students realize this as a cooperative venture - one day they may be the designer/coder, the next they may be the "naive" user. They expend considerable effort making certain their program executes properly and is designed for human use. Not entirely by accident, they are learning the components of good computer-based education lessons, both as a designer and a user.

4. Use of cooperative exercises, computer and otherwise

Until relatively recently, this topic would have been virtually nil. However, there appears to be an increasing awareness that a model of cooperation is at least as important for education as that of competition. Star Trek, Moonwar, Battleship, etc., all have a certain fascination, but how about cooperative games also? In a recent issue of Computers and Society, Bruce Hicks discussed several such games [10]. "SOS" is available on PLATO and "Lost and Forgotten Island" is available in Basic. In these games, cooperation is espoused for the mutual benefit of all players. After playing these, students (and instructors) now have some models for community-building lessons. There have also been several programs in issues of Creative Computing (a great magazine!) which promote cooperation rather than shooting down Klingons [3].

I do not restrict cooperation to be viable only when interacting with the computer. This is, however, an increasingly important type of CBE lesson, in my opinion. Students also share together in discussion about films. Controversial ones such as The Right of Privacy or A Matter of Survival (both available from Indiana University) prompt conversation in which significant and honest feelings are exchanged. People learn to relate to others who may have strong, but different, thoughts. Many of the exercises in Values Clarification by Simon, Howe, and Kirschenbaum are useful in this domain also [22].

Recently, Judy Edwards, new president of the Association for Educational Data Systems (AEDS) brought to my attention the BLUE/GREEN classroom simulation dealing with cooperation within a company framework. This type of activity is growing in popularity as witnessed by the recent formation of the New Games Foundation. Quoting from the director, Pat Farrington, we have:

> "If people center on the joy of playing, cooperating and trusting rather than striving to win, they become part of the process - not spectators ... We try to create individual responsibility -- helping people create their own games ... We're into cooperating rather than competing." [7]

It would appear that more and more people are realizing that cooperation is an increasingly vital element in our lives. That is why I am concerned that students have first-hand experience with this model.

5. Parties

One of the essential ingredients of community is knowing one another. Parties provide an excellent first step for learning one another on an informal and non-threatening basis. And, because they are so much fun, we try to have several!

The "content excuse" for most parties is the celebration of a student's birthday. Donuts and cider usually provide a common basis for sharing physical sustenance while the class continues. Frequently the student may indicate their special plans for the day.

The date of birth is ascertained from the first program taken by the students. An interview program Query, written in Basic, asks them for their favorite food,and tries to identify them by their preference in later interviews. It then asks them for their birthday, and goes on to ask if they have any questions. The latter feature is used as a note/comment/question file throughout the course. The student has the option of refusing to answer any question, e.g., for privacy reasons. She/he also has the option of terminating the interview and thus having no record kept.

These informal in-class celebrations allow us to learn a little more about a person than simply what courses they are taking. They are also supplemented by a get-together every semester, often at the instructor's home. Each person brings their appetizer specialty, and the evening is spent with good food and informal conversation. The latter often arises from a game that each person brings, as well as programs and games available over the interactive terminal I bring to the party. Again, people put out individual efforts to build for common goals and mutual benefit. Cooperation is not an impossible model. It is, I believe, a most possible, desirable, and necessary one for any world in which human beings form a part.

Summary and Future

In summary, I have tried to indicate why I believe cooperation to be a valuable and integral part of education. It is compared to a traditional mode, usually based on competition, and then to the more recent mode of individualized instruction. The latter has often been implemented, in my courses and others, within a PSI environment and frequently involves computer-based education. These individualized techniques have done much to eliminate the idea that only 10% of students are able to master the material presented. However, most serious problems in the world are solved by a community of responsible people. For just one example, Howard Casmey, in the address referenced earlier, dwells at length on the necessity of partnership (cooperation) between companies and education for gaining widespread acceptance of quality CBE.

It is in this belief that cooperation is introduced explicitly in my introductory computer science courses. The primary cooperative efforts are:

1) study partners for PSI units,
2) design of a computerized personal data base by a small group,
3) mutual design and use of interactive programs,
4) use of cooperative exercises, computers and otherwise, and
5) parties.

I am under no illusion that these are the only or best ways to introduce cooperation explicitly. Other techniques with which I am familiar include: computer clubs for hobbyists and micro-computer users, computer uses for elders (another project of Hicks) [8], designing programs for the school itself or local community, and, of course, the very widespread efforts in primary and secondary education. Learning exchanges, open classrooms, individually-guided education (IGE) programs, etc., are all oriented toward subject mastery. They also usually occur in a self-paced environment and often involve interpersonal exchanges [5]. I personally would be very grateful if the reader would share additional references for cooperative efforts, games, programs, etc.

What of the future? Only the future itself knows. But I would hope that my understanding of people and the learning process would grow and mature. It may well be that cooperative efforts are "not all they're cracked up to be." The values of one's own self and of one's God are probably pre-eminent. Yet at the same time, we live, love, work, and learn in a community of others. I am grateful for that opportunity, enjoy helping others, and enjoy being helped by a caring community of responsible individuals.

APPENDIX

Group Project

DATA BASE DESIGN
FOR
PERSONAL INFORMATION

CS 220
Ken Modesitt

> I KNOW EVERYBODY'S INCOME AND WHAT EVERYBODY EARNS;
> AND I CAREFULLY COMPARE IT WITH THE INCOME-TAX RETURNS;
>
> TO EVERYBODY'S PREJUDICE I KNOW A THING OR TWO;
> I CAN TELL A WOMAN'S AGE IN HALF A MINUTE - AND I DO!
>
> YET EVERYBODY SAYS I AM A DISAGREEABLE MAN!
> AND I CAN'T THINK WHY!
>
> KING GAMA IN GILBERT AND SULLIVAN'S
> PRINCESS IDA

There is today a vast array of data about you and me stored in computerized data bases. The following list, although very incomplete, contains types of organizations and institutions which can input and access data about you from thousands of computers throughout the nation.

- Federal Government
  - Census Bureau
  - FBI
  - IRS
  - Selective Service
  - Armed Forces
  - Social Security Administration
  - State Department (Passports)

- State Government
  - Police
  - Department of Motor Vehicles
  - Department of Revenue

- County Government
  - Tax Assessor
  - Board of Elections

- City Government

- Banks and Savings and Loans
- Educational
- Insurance
- Utilities
- Employer
- Credit Cards
- Professional Organizations

Group Project
CS 220
Ken Modesitt

Special Interest Groups
  - Alumni
  - NRA
  - John Birch Society
  - Book Clubs
  - Record Clubs
  - League of Women Voters
  - Audubon Society
  - etc.

Medical Records (Physicians, hospitals)
Mail-order Houses
Political Parties
Dating and Marriage Services
Retail Stores

The potential good from wise and careful use of such automated personal data systems is considerable. And so is the potential harm! I am particularly concerned about the latter. Incidents, some humorous and some serious, abound which relate to computerized personal information systems. Can you think of several?

In response to a growing concern about the role of the individual in such computerized systems, the United States Department of Health, Education, and Welfare published a special report. "Records, Computers, and the Rights of Citizens" (1973) is on reserve in the library. The report recommended the enactment of a Federal "Code of Fair Information Practice." The Code rests on five basic principles that would be given legal effect as "safeguard requirements" for automated personal data systems. The following is taken verbatim from the report (pp. 40-41).

> "HERE THEN IS THE NUB OF THE MATTER. PERSONAL PRIVACY, AS IT RELATES TO PERSONAL-DATA RECORD KEEPING MUST BE UNDERSTOOD IN TERMS OF A CONCEPT OF MUTUALITY. ACCORDINGLY, WE OFFER THE FOLLOWING FORMULATION:
>
> > AN INDIVIDUAL'S PERSONAL PRIVACY IS DIRECTLY AFFECTED BY THE KIND OF DISCLOSURE AND USE MADE OF IDENTIFIABLE INFORMATION ABOUT HIM IN A RECORD. A RECORD CONTAINING INFORMATION ABOUT AN INDIVIDUAL IN IDENTIFIABLE FORM MUST, THEREFORE, BE GOVERNED BY PROCEDURES THAT AFFORD THE INDIVIDUAL A RIGHT TO PARTICIPATE IN DECIDING WHAT THE CONTENT OF THE RECORD WILL BE, AND WHAT DISCLOSURE AND USE WILL BE MADE OF THE IDENTIFIABLE INFORMATION IN IT. ANY RECORDING, DISCLOSURE, AND USE OF IDENTIFIABLE PERSONAL INFORMATION NOT GOVERNED BY SUCH PROCEDURES MUST BE PRESCRIBED AS AN UNFAIR INFORMATION PRACTICE UNLESS SUCH RECORDING, DISCLOSURE OR USE IS SPECIFICALLY AUTHORIZED BY LAW.

Ken Modesitt
Group Project
CS 220

> THIS FORMULATION DOES NOT PROVIDE THE BASIS FOR DETERMINING A PRIORITY WHICH DATA SHOULD OR MAY BE RECORDED AND USED, OR WHY, AND WHEN. IT DOES, HOWEVER, PROVIDE A BASIS FOR ESTABLISHING PROCEDURES THAT ASSURE THE INDIVIDUAL A RIGHT TO PARTICIPATE IN A MEANINGFUL WAY IN DECISIONS ABOUT WHAT GOES INTO RECORDS ABOUT HIM AND HOW THAT INFORMATION SHALL BE USED.
>
> SAFEGUARDS FOR PERSONAL PRIVACY BASED ON OUR CONCEPT OF MUTUALITY IN RECORD-KEEPING WOULD REQUIRE ADHERENCE BY RECORD-KEEPING ORGANIZATIONS TO CERTAIN FUNDAMENTAL PRINCIPLES OF FAIR INFORMATION PRACTICE.
>
>> THERE MUST BE NO PERSONAL-DATA RECORD-KEEPING SYSTEMS WHOSE VERY EXISTENCE IS SECRET.
>>
>> THERE MUST BE A WAY FOR AN INDIVIDUAL TO FIND OUT WHAT INFORMATION ABOUT HIM IS IN A RECORD AND HOW IT IS USED.
>>
>> THERE MUST BE A WAY FOR AN INDIVIDUAL TO PREVENT INFORMATION ABOUT HIM OBTAINED FOR ONE PURPOSE FROM BEING USED OR MADE AVAILABLE FOR OTHER PURPOSES WITHOUT HIS CONSENT.
>>
>> THERE MUST BR A WAY FOR AN INDIVIDUAL TO CORRECT OR AMEND A RECORD OF IDENTIFIABLE INFORMATION ABOUT HIM.
>>
>> ANY ORGANIZATION CREATING, MAINTAINING, USING, OR DISSEMINATING RECORDS OF IDENTIFIABLE PERSONAL DATA MUST ASSURE THE RELIABILITY OF THE DATA FOR THEIR INTENDED USE AND MUST TAKE REASONABLE PRECAUTIONS TO PREVENT MISUSE OF THE DATA."

Your cooperative group assignment is to design an automated personal data system for a content area of your choice. Feel free to use those listed earlier. Your design should include the principles just articulated and any others you would like.

What features would you want in such a system were you a user and/or entry thereof?

Would you want to see your record?

What information can others input about you?

Can you challenge and change your record? How?

Who has looked at your record? When?

Who can erase information?

Group Project
CS 220
Ken Modesitt

Can you prevent someone from looking?

How do you know incorrect or outdated information was really erased (corrected)?

Is the information used only for that purpose for which it was collected?

How would you find out if a record of you is being kept?

Are there levels of information access?

Can you inspect the records of your mate, friend, children, employer, etc.?

Is certain information purged automatically after a certain period of time?

The above questions represent only a very few which could be asked. You will undoubtedly uncover others as your project unfolds. You might ask others what questions they have.

The project is a design one only; it does not involve coding (unless you wish to). However, the design should proceed in a top-down manner, expanding various steps where necessary. The final written result of your effort should be a set of structured programs given in increasing detail. The most detailed one should be one able to be coded directly into some programming language.

A typed copy of your design will be presented to class during the last two weeks of class (I will supply the mimeograph service if you give me the typed copy 2 days before handing it out.). In addition, your presentation for that day of class may take any form you wish, as long as each group member has a role. Your group might role-play a "typical day in the life of a computerized dating service," for example. Or someone might indicate how a user might try to use/access your system, honestly or otherwise. Perhaps you could incorporate members of the class in your presentation. In any case, you are to hand in two questions (and their answers) which could be answered only by a person attending your presentation. Some of these will appear on the final exam.

Good luck and hope you enjoy this "realistic" assignment involving the world and computers and other people!

Ken

P.S. - Another excellent reference is Privacy Journal. I have several copies in my office.

BIBLIOGRAPHY

1. Casmey, Howard B. "Computer-Based Education: An Approach Toward Adaptive Learning Procedures." ADCIS Summer Conference, 1976, pp. 1-22.

2. Cheney, Paul. "Egoless Programming: As an Instructional Technique." AEDS 14th Annual Convention, 1976, pp. 222-224.

3. Creative Computing, David Ahl (editor), Morristown, New Jersey.

4. Egerton, John. "Teaching Learning While Learning to Teach." Change, Volume 8, Number 2, March, 1976, pp. 58-61.

5. Gehret, Kenneth G. "Children Make the Best Tutors." Christian Science Monitor, January 14, 1974, p. F1.

6. Harman, Willis W. An Incomplete Guide to the Future, Stanford Alumni Association, Stanford, California, 1976.

7. Harrington, Pat. "'Earth Ball' Rolls into America." Fort Wayne Journal-Gazette, Fort Wayne, Indiana, February 15, 1976, p. 2D.

8. Hicks, Bruce and Kathy Jaycox. "Elders, Students and Computers: A New Team." ISEAC #7, College of Education, University of Illinois, Urbana, Illinois, 1976.

9. Hicks, Bruce and Zielinski. "SEC-A Simulation of Computer Privacy and Security Problems." ISEAC #15, College of Education, University of Illinois, Urbana, Illinois, 1976.

10. Hicks, Bruce. "Computer Outreach." Computers and Society, ACM Special Interest Group, Volume 7, Number 3, Fall, 1976, pp. 10-14.

11. Journal of Personalized Instruction. J. Gilmour Shermen (Editor-in-Chief), Center for Personalized Instruction, Georgetown University, Washington, D.C.

12. Lincoln, C. Eric. "Equalizing the Opportunity to Learn." Change, Volume 8, Number 6, July, 1976, pp. 60-63.

13. Modesitt, Ken. "The Tangled Triangle: Cooperation, Computer-Based Education and Personalized Systems of Instruction." ADCIS Summer Conference, 1976, pp. 320-324.

14. Modesitt, Ken. "Personalized Systems of Instruction in Computer Science: 'Adult' Education." National Conference on Personalized Instruction in Higher Education, 1975.

15. Modesitt, Ken. "An Excellent Mixture for PSI: Computer Science, PLATO, and Knowledge Levels." ACM National Conference, 1974, pp. 89-94.

16. Modesitt, Ken. "PSI: A Valuable Addition to the Alphabet Soup for Computer Science Education." ACM Special Interest Group on Computer Science Education Bulletin, Volume 6, June, 1974, pp. 37-44.

17. Morris, William (editor). _The American Heritage Dictionary of the English Language_, 1969.

18. North, Robert C. _The World that Could Be_. Stanford Alumni Press, Stanford, California, 1976.

19. _PSI Newsletter_. Center for Personalized Instruction, Georgetown University, Washington, D.C.

20. Rhinelander, Philip H. _Is Man Incomprehensible to Man_? Stanford Alumni Press, Stanford, California, 1973.

21. Ruskin, Robert S. and Steven F. Bono (editors). _Proceedings of the National Conference on Personalized Instruction in Higher Education_. Georgetown University, Washington, D.C., 1974 to present.

22. Simon, S. B., L. W. Howe, and H. Kirschenbaum. _Values Calrification: A Handbook of Practical Strategies for Teachers and Students_. Hart Publishing Co., New York, 1972.

THE COMPUTER DISPLAY AS A MEDIUM IN THE TEACHING OF AESTHETICS IN VISUAL DESIGN

Raymond D. Nichols
Department of Art
University of Delaware
Newark, Delaware 19711

and

James H. Wilson
Delaware PLATO Project
University of Delaware
Newark, Delaware 19711

ABSTRACT

Computer graphics have been investigated and improved markedly in recent years, but their application to art education has been largely neglected. In order to facilitate instruction in an introductory course in basic design, programs were developed on the PLATO system to allow computer graphics to serve as a medium for a student activity that had previously been done using traditional media.

This use of the computer for the execution of technical procedures was aimed at three educational goals: 1) Students, able to revise previous work with minimal effort, are less likely to alter their aesthetic judgment as a result of effort expended than has been the case using traditional media. 2) Students and faculty, no longer influenced by variations in the students' technical abilities, are forced to concentrate on aspects of visual design. 3) Students using the program should be encouraged by it to involve thought processes throughout the experience, rather than to divorce creative thought, execution, and evaluation.

The courseware has been used by 60 University of Delaware freshmen, and preliminary results show that initial goals were achieved. Further, students were able to execute more intricate designs in a shorter time. Finally, students profited from a greater opportunity for ongoing feedback, both through interaction with their instructor and through viewing of classmates' designs. These factors have combined to emphasize perception of the experience as education rather than simply production.

Instructional Environment

Basic design is traditionally taught in fine arts departments of our universities and colleges, usually in conjunction with basic drawing, and serving as a foundation for the fine and applied art curricula. These basic design courses are usually designed with three main goals or responsibilities toward the art student's education:

1) establishment of a familiarity with various art media
2) development of basic hand skills in manipulating tools associated with the art process
3) development of a proficiency with the thought process in relation to design problem solving

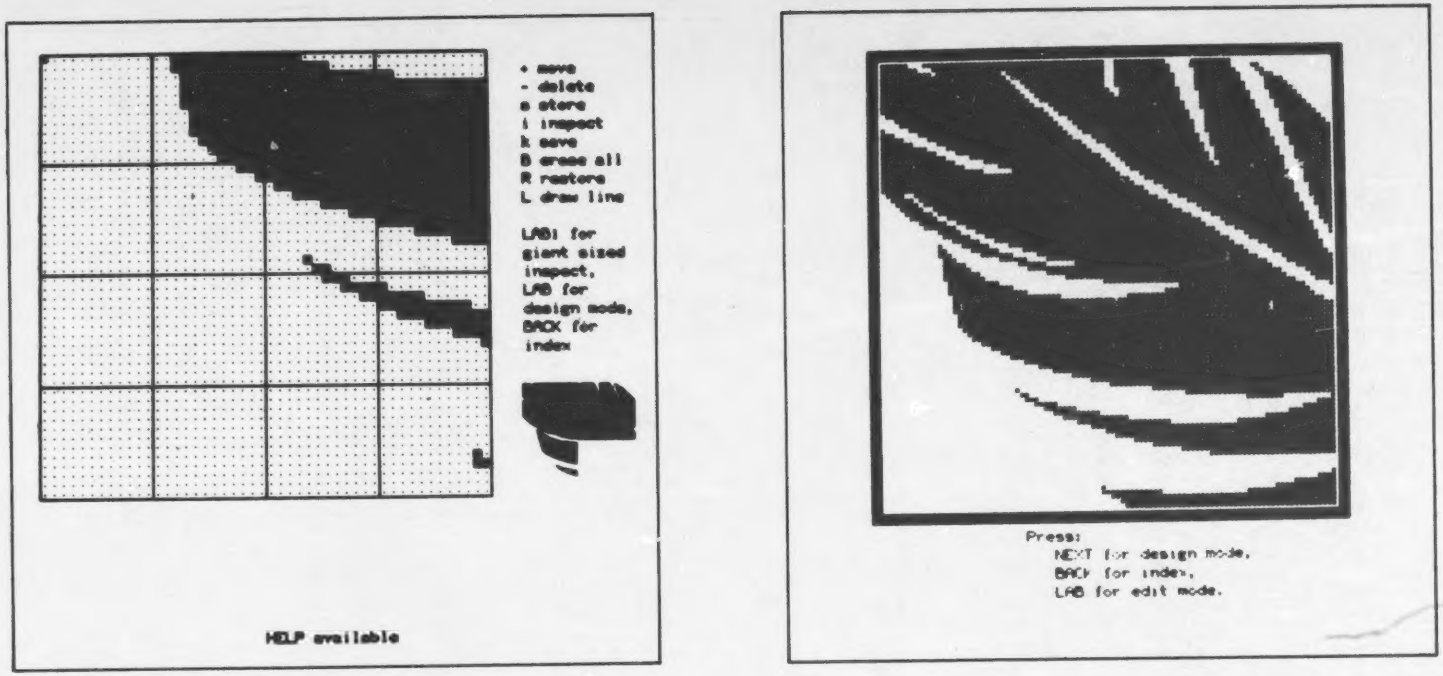
These design courses are usually taught through a series of problems, dealing with aesthetic or utilitarian functions, in which the students plan and execute a solution to the problem. According to Koberg and Bagnall (1972), the problem solving process that the student should utilize can be reduced to seven steps:

1) acceptance of the situation
2) analysis of the general problem
3) definition of the specific problem to be solved
4) ideation
5) selection
6) implementation
7) evaluation of the results

If we analyze the three course goals in conjunction with the seven problem solving steps, we can see that familiarity with media is necessary only at step five with selection from the ideas, and hand skills are necessary only at step six with the implementation of the student's ideas. The general problem that has been observed in the classroom is a general lack of ability on the part of our art students to accomplish adequately steps two to four, which significantly lessens the possibility of any consistency at design problem solving.

It is therefore a goal in the presentation of lessons utilizing computer-assisted instruction to attempt to isolate the thought process from the technical process (or at least significantly reduce the emphasis on the technical process), thus allowing more stress on the steps dealing with analysis, definition, and ideation. This approach would:

1) reduce students' confusion in separating mental and technical skills
2) reduce students' preconceived notions about media and tools, focusing new thinking about the possible solutions
3) allow for a less biased and more understandable comparative evaluation of students' efforts by reducing the level of technical skills required for the problem to that attainable by all students


+ move
- delete
s store
i inspect
k save
B erase all
R restore
L draw line
LAB1 for giant sized inspect,
LAB for design mode,
BACK for index
HELP available
Press:
NEXT for design mode,
BACK for index,
LAB for edit mode.

a

b

Figure 1. Sample pages from the lesson, showing (a) the portion in which the lower left quadrant of the unit is enlarged (8x) and edited, and (b) the completed unit, enlarged (4x).

This presentation has a further advantage as applied to a major problem with educational assignments involved in aesthetic evaluations: whether the aesthetic dictates the design (which we would prefer) or whether the design dictates the aesthetic. The problem becomes acute when both the technical skills and the mental process are being used together. Students begin an effort with certain aesthetic goals and the requirements necessary to the development of an aesthetically pleasing design in mind. They then put considerable effort into the project. Probably at some point as they work, a need arises to alter work which has already been completed in order to remain consistent with aesthetic values. What traditionally happens here, unfortunately, is this: Instead of redoing parts of previous work, they simply alter aesthetic values, consciously or unconsciously, to fit the finished design. The tendency toward such an action is at odds with the development of the desired mental process. A CAI lesson should effectively reduce the effort of the execution once the original unit has been input. In such an environment, students are encouraged to alter the design to conform to their aesthetic values.

Lesson Design

CAI lessons in art could utilize a variety of modes of instruction. Here, we have concentrated on a form that might be considered similar in some ways to simulation or problem solving, but in other ways is unique.

Given the capabilities of a powerful graphics terminal, possibilities for exploration by creative students could easily prove too numerous to be manageable for purposes of instruction in design. Instead, various structures have been imposed by several developers to provide a common ground for consideration of designs according to selected relevant characteristics.

One example of an imposed structure is provided by programs in which a complex line drawing is generated on the basis of numbers or functions entered by the student. A characteristic of most of these programs is that very few of their users can visualize the outcome at the point at which the determining numbers or functions are chosen, and thus the only process typically followed is selection. Although many of the resultant designs are aesthetically appealing, the programs may tend to focus more attention on the mathematical relationships involved than the process of design.

Other CAI procedures allow a student more easily accessible control (and less surprise) over the product. A simple and elegant example of such a lesson is "Touchpaint," by Priscilla Obertino and Jim Gilfillan. In it, the student simply touches an area of the computer display to have it outlined by a triangle. Successive touches to the same area fill it in and return it to its original state. With such an easily understandable procedure, users from the primary level upwards can produce designs of a moderate level of sophistication.

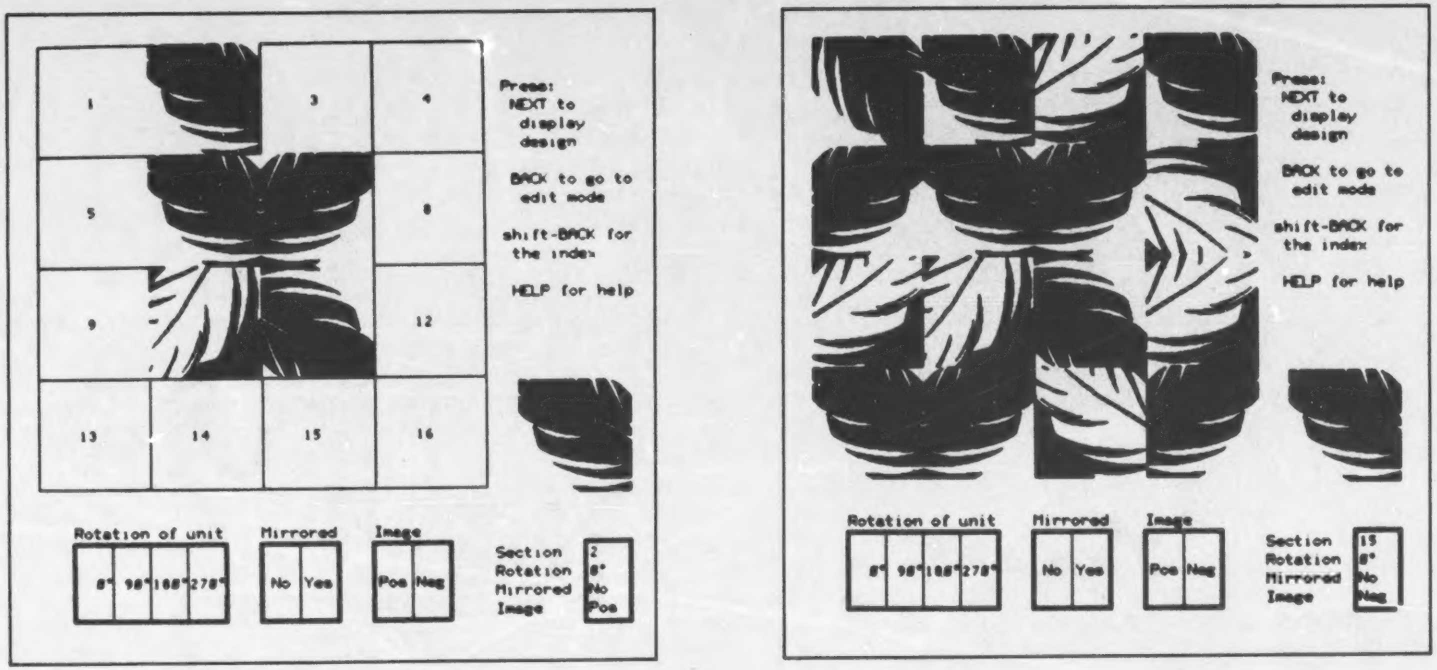
Another procedure is to present a standardized format, such as a pattern of blocks, each containing several sizes of squares (as used by Kolomyjec (1975)) and allow the student a series of choices as to such variables as number of squares, centering, and rotation.

Our main efforts here concern a combination of these approaches, as applied in a basic design course at the University of Delaware. The "Unit Design Program" provides for the designing of a two-dimensional image (called a "unit") which can be placed into a 4 x 4 array of such images by rotating, mirroring and/or inverting the positive/negative relationships of each element. The problem is begun with the execution of a drawing on paper covering an 8" x 8" area. Objects of drawings can be specified as organic, man-made, or of other classes of an instructor's choice. The student then isolates a 4" x 4" area of the drawing and transfers this design to a 96 x 96 sheet of graph paper (normally 1/5" divisions) which is then used to input the "unit" into the computer. The image of the "unit" consists of the dot-like elements of a plasma panel, each of which can be on or off, combining to create the desired image. As these dots are in a fixed position, the method for inputing a given shape is the same for all students, reducing the technique of physically drawing the image to a simple transferring procedure. (See Figure 1.) The student then progresses into design mode, where this "unit" is used to create a modular design comprised of a 4 x 4 array of "units." In this mode the student can alter the "unit" by rotating 0°, 90°, 180°, or 270°, mirroring from any of those positions, and/or having the positive/negative relationship reversed. The modular design is worked on until the student arrives at what is felt to be the best, or a series of the best, of the available solutions with regard to individual aesthetic tastes. These designs can then be judged for aesthetic value in relation to each other. (See Figure 2.)

Evaluation

The lesson was used by 60 University of Delaware freshmen during Fall semester, 1976. Students completed an attitude survey, and qualitative assessments of student progress were done by the instructors as well as fellow students.

Most students indicated that they experienced the following feelings "often" during their work with the PLATO lesson: fun, involved, encouraged, challenged, in control, successful. Most students "seldom" felt bored or annoyed. Inconclusive results were obtained on the feelings "frustrated" and "confused," though responses tended more toward "seldom" than "often." Importantly, students felt that the mechanics of using the PLATO system did


1
3
4
5
8
9
12
13
14
15
16
Rotation of unit
0°
90°
180°
270°
Mirrored
No
Yes
Image
Pos
Neg

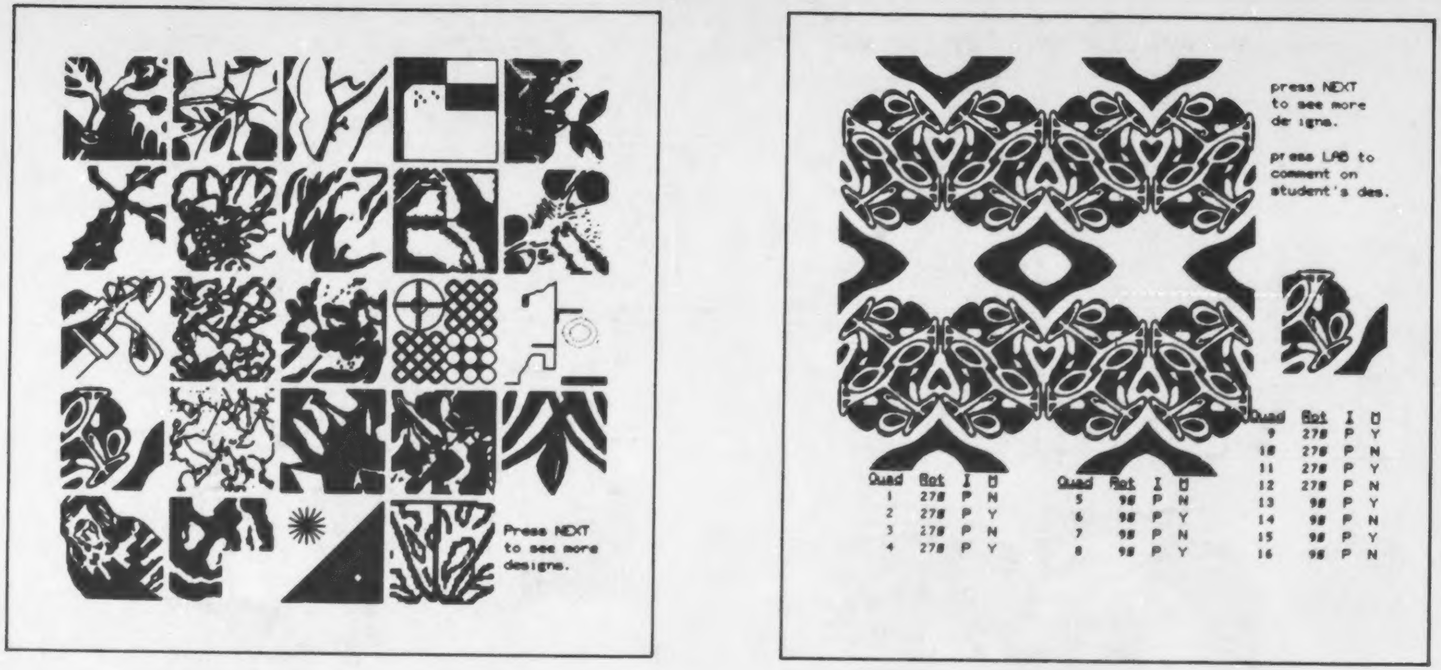
Press:
NEXT to display design
BACK to go to edit mode
shift-BACK for the index
HELP for help

Section 2
Rotation 0°
Mirrored No
Image Pos

Press:
NEXT to display design
BACK to go to edit mode
shift-BACK for the index
HELP for help

Section 15
Rotation 0°
Mirrored No
Image Neg

Figure 2. Sample pages from the lesson's "design mode," in which students combine images of the unit, possibly altered as to rotation, mirroring, and positive/negative inversion.



Press NEXT
to see more
designs.

press NEXT
to see more
designs.

press LAB to
comment on
student's des.

```
Quad   Rot   I   U          Quad   Rot   I   U          Quad   Rot   I   U
   1   270   P   N             5    90   P   N             9   270   P   Y
   2   270   P   Y             6    90   P   Y            10   270   P   N
   3   270   P   N             7    90   P   N            11   270   P   Y
   4   270   P   Y             8    90   P   Y            12   270   P   N
                                                          13    90   P   Y
                                                          14    90   P   N
                                                          15    90   P   Y
                                                          16    90   P   N
```

a

b

Figure 3. Displays available to instructors, showing (a) 24 units, and (b) a student's completed design, showing choices made for each sector.

not detract from their learning.

Student progress in the PLATO lesson differed both quantitatively and qualitatively. To produce a design on PLATO of the same degree of complexity as another done by traditional media took one third to one half the time. As a result, the designs done by computer were generally more intricate. Further, many of the designs exhibited a unity of concept in the interactions of the units in the modular design, perhaps partially attributable to the ease of modifying previous work. A pleasant additional result noted was the interactive and communicative atmosphere among students working in a CAI classroom atmosphere.

Sample units generated by the program are shown in figure 3. Additional slides will illustrate other aspects of student work.

ACKNOWLEDGMENT

The authors wish to express great appreciation for the efforts of Charles Wickham, who spent many hours on the programming of the unit design problem.

REFERENCES

Koberg, Don and Bagnall, Jim. The Universal Traveler. William Kaufman, Inc., Los Altos, Ca: 1972.

Kolomyjec, William J. "The Appeal of Computer Graphics." In Ruth Leavitt (Ed.), Artist and Computer. Harmony Books, Inc. New York: 1976.

Comprehensive Computer Based Education Development in a Network Environment

by

James E. Eisele

John S. Edwards

INTRODUCTION

By now everyone knows that actual utilization of computers in education falls woefully short of the enthusiastic expectations of a decade, or so, ago. Furthermore, we have learned a great deal about why the potential of computers in education has not been fully realized. This paper describes an approach to computer based education (CBE) which seems to facilitate a rapidly growing utilization of computers in instruction.

The program being described is one and one-half years old. Initial efforts to implement CBE in the Georgia University System Computer Network began in the fall of 1975. Progress has been rapid; acceptance by faculty members in the Network has been widespread. We feel that the growth of CBE in the Network is attributable in part to the readiness of faculty to try something new, but also in part to the approach used in development of the program.

BACKGROUND

The University System (of Georgia) Computer Network (USCN) is composed of thirty-four institutions which share a Control Data CYBER 70, Model 74. The USCN began in 1970 with the assistance of a grant from the National Science Foundation. Initially, each site functioned independently in the Network with monthly meetings for status reporting. In 1975 the Network was subdivided into six regions with representatives from each forming an Executive Committee. Currently, the Executive Committee provides liason between the Network sites and the Central Site.

The Central Site houses the CYBER 70 and two IBM 370/158's. There are four functional divisions at the Central Site, one of which has major responsibility for educational and technical support services for the Network. This Division is composed of personnel in systems programming (for the CYBER), user services, and educational applications. All these personnel are available for site visits, although there is a designated traveling consultant working out of the User Services Department.

The Division of Educational and Technical Support is dedicated to the goal of improving the effectiveness and utilization of computer-based education. In the past two years, the USCN has progressed from the point of near zero level instructional computing to a high degree of activity in the area. Research computing in the Network has also shifted from total reliance on batch processing to moderate use of interactive applications.

The Division is managed by personnel who have a background in, and understanding of, educational innovation, and is staffed by persons with technical competency. This combination of personnel is a significant factor involved in the approach to CBE development which has been employed. The variety of personnel expertise has established conditions which encourage CBE adoption by Network faculty and has circumvented conditions previously identified as inhibiting CBE utilization.

In addition to appropriate staffing, successful implementation has been enhanced by utilizing principles of curriculum development and instructional design that have long been accepted in practice, if not in theory. Briefly, these principles consist of the following:

1 - Curriculum development is a continuous, on-going process;

2 - Curriculum development should begin with real, existing, instructional problems;

3 - Curriculum development should focus on many subject areas simultaneously rather than pursue a piece-meal approach;

4 - Curriculum development should include the participation of those who will implement the results;

5 - Instructional design should follow along the lines of the learning cycle;

6 - Instructional design should rely upon the expertise of appropriate personnel for making decisions about content, learners, learning, and technology.

These principles provided the underlying philosophy of the new Division within the existing computer network. The following description of the program should be viewed according to these guidelines.

GOALS

The general goal of the program is to provide support and leadership for all educational users of computer technology in the University System of Georgia. This goal includes providing assistance in all phases of educational computing - from stimulating greater and improved utilization to providing "crisis intervention" for routine users. More specifically, the goals of this program include:

1. To motivate educational personnel to increase and improve their utilization of computers as a major tool for education.
2. To provide courseware and software for educational applications.
3. To promote the development of curricula and lessons for computer-based educational systems.

4. To assist educational personnel in the acquisition of competencies necessary to utilize the computer as an educational tool.

5. To provide educational and training opportunities for users and supportive personnel of computer-based educational services.

6. To develop innovative applications of computers in education.

7. To provide technical assistance for all routine functions of computers used in the educational environment.

8. To assist in the evaluation of educational applications of computer technology.

OBJECTIVES

In order to accomplish these goals, some means of practical implementation had to be developed. Therefore, objectives were defined in behavioral terms with a series of activities expected to evolve from the definitions. For purposes of analysis, a matrix was constructed to plot categories of CBE applications against specific functions which are an integral part of any educational process. (See Figure 1). The categories of CBE applications were defined by the National Center for Educational Statistics' (NCES) Handbook:

1. Computer-Assisted Instruction
2. Computer-Managed Instruction
3. Computer-Developed Instruction
4. Computer-Assisted Testing
5. Computer-Assisted Guidance

The specific educational functions used in our analysis include:

1) Diagnosis of Learner's Needs
2) Management of Instructional Objectives
3) Design and Implementation of Teaching Strategies
4) Coordination of Educational Media
5) Monitoring and Measuring Learner Achievement and Instructional Effectiveness
6) Generalized Management
7) Research and Development

The matrix was used as a planning tool to develop Divisional activities and to allocate resources and establish priorities. This device has proved quite effective. Needs expressed by our network faculty have been described well by the columns (educational functions). Instructional support requirements identified in the University System have not yet required additional functions.

PROCEDURES

The process of deriving appropriate objectives is one of the primary activities in which we participate and provide leadership. In fact, this activity involves considerable time and effort to interact with diverse groups of users and to assist with the definition of objectives which are important to the users. These activities can be categorized as coordination functions. When the objectives are defined and clarified, we find the functions to be systems development and installation, demonstration and training of users, and continued technical support for system maintenance and upgrading.

Network Coordination

All activities which support CBE from the Central Site and from within the Network are coordinated through representatives of the Network. Monthly meetings are held by the Network Executive Committee within the six regions of the State, an annual meeting is held for all Campus Coordinators, and meetings are called about twice a year by Central Site staff for all Campus Coordinators. All of these sessions serve the purpose of presenting current developments, discussing additional needs, setting priorities, clarifying Central Site activities, and making other decisions relative to development activities - such as who will do the work, what is to be done, and who will provide advice and support.

A Network travelling consultant, who function much as a circuit-riding preacher, coordinates visits to campus sites for individual discussion of local needs within the Network. Though a single individual is designated as the circuit rider the actual visits may be made by whichever staff member can provide the greatest amount of assistance. In all cases where CBE applications are the primary topic of discussion, a staff member whose expertise is in educational applications, rather than computer technology, usually makes the visit.

In addition to coordination with the Network, the Central Site staff is coordinated with the Academic Advisory Committees of the Board of Regents. These committees are composed of faculty representatives of each academic area from each campus. We have not yet finalized arrangements with all committees but it appears that they will provide a valuable advisory board for evaluating the content of computer applications and will assist in making curricular decisions pertaining to adoption of applications.

Systems Development Installation

A major function of the Division involves selecting, evaluating, and acquiring existing courseware and software for educational applications. During the testing phase, the documentation of reported applications is reviewed for completeness and accuracy; frequently, revision of both the software and documentation is required before the material can be released for general use. Occasionally, the content of the courseware also requires editing; academic departments in the University System are consulted whenever possible for this function.

The Division provides leadership for creating both courseware and software for use in all academic disciplines and at all institutions of the network. This involves assisting faculty members from various institutions to author new curriculum and instructional materials, and to create software whenever necessary. In return for this assistance, the materials are added to the network library for general use when applicable. In this way, we will be able to build a library of useful CBE packages which relfect the unique requirements of our University System. Every attempt is made to assure that authors of new materials are given appropriate consideration and recognition for their efforts in carrying out this function.

Currently, three major applications are functional. The first application, CAI libraries, was preceded by an extensive search for CAI courseware which was operational. Acquisition of existing CAI material was selected as the initial implementation for three basic reasons:

1. The central site staffing was proceeding slowly and deliberately, therefore, we were not able to allocate a great deal of personnel resources to CBE.
2. The immediate presence of a library of operational CAI lessons would provide a source for demonstrating the advantages of this instructional mode.
3. The extent to which faculty would use these libraries would provide some preliminary indications of the prevailing general attitude towards CBE.

The System faculty has responded to these materials in a very positive manner. The use has grown from fifty to fifteen hundred accesses per month. Furthermore, faculty are submitting to us lessons which have been used on local campuses, thereby allowing system-wide availability. We have noticed that this sharing of materials has had a beneficial effect both on instructional improvement and on CBE development. Apparently the CAI lessons are a focal point around which an intra-system professional comaraderie is developing. Authors are quite responsive to their colleagues' suggestions for editing lessons, and from this type of dialogue a consensus of instructional intent and content seems to be evolving. Most of the lessons are written in the BASIC language; as more users learn BASIC we expect the libraries to expand further. Currently there are about four hundred lessons in ten subject areas.

The second application to be installed was a computer-assisted test construction system. A search of available systems revealed that either the programming languages were incompatible with our hardware or the design did not satisfy our requirements. Consequently, we developed our own system from vendor utilities with local design and programming. The locally created modules are programmed in Fortran and run on the CYBER 70/74. The system incorporates elements of interactive and batch processing. Briefly stated, the following major functions are available:

1. Data base management (input and editing capability) for test item files
2. A conversational mode interactive module for searching the item files and printing test items
3. Test and item analysis reports

The TCS currently has sixteen data bases from as many subject areas. The interactive search of these files allows the user to specify four parameters: content category (a type of index), difficulty level, cognitive level, and type of test item (multiple-choice, true-false, essay, or short answer). The user also specifies the quantity of items desired.

This system reached operational status in November of 1976. Since that time, it has received a very warm reception. During November and December, faculty in fifteen institutions began regular use for generating classroom examinations.

In addition, several departments are investigating the feasibility of generating departmental entrance and/or exit examinations. Particular interest has been shown in developing local norms and in validating test content for the University System required curriculum.

In November, the School of Veterinary Medicine approached our Division for help in managing a very large compendium of competency statements. This request merged exactly with our plan to implement an objectives management system. Consequently, we have used the Veterinary statements as a model for developing the initial phase of our Computer Managed Instruction (CMI) System.

The functions of the CMI system will consist of data entry procedures, file creation, editing and maintenance programs, and report generating programs to print indexes. The computer file is data element oriented which allows the flexibility required for such complicated data management.

The software for handling this project has already been written. The file building and maintenance programs are the University of Georgia Data Base Generation System. The report generator was written by Pier-Paolo Pisani of the Office of Computing Activities, University of Georgia.

The data element orientation of the file structure was chosen for two reasons:

1. To permit an open-ended file construction process. This technique allows unlimited addition to each file record as new data becomes available.

2. To allow use of an indexing model for storing and retrieving behavioral objectives.

The data element structure may be explained as a logical record which consists of an unlimited quantity of discrete units (data elements). The logical record may be thought of as the competency statement plus all pertinent data which relate to the statement. In the initial development, the Magerian objective model was used to structure the statements. This model consists of the statement of the task, the statement of the conditions under which the task is performed, and the criteria for evaluating the student's performance on the task. The choice of the Magerian model will not restrict the subsequent use of other models for defining competencies in other disciplines. The data element oriented file structure can be modified by creation of new data elements at anytime. Thus, if a different model is preferred during subsequent development, the file can accommodate this by the creation of appropriate new data elements.

The data elements generally fall into two classes:

1. Active type - which are elements that contain knowledge to acquire, tasks to perform, evaluations, etc.

2. Passive type - which contains descriptions of prerequisite skills and knowledge, resource indicators, cross-references to related objectives, etc.

The flexibility and open-ended file structure will allow maximum potential for accommodating additional information as suggested by subsequent investigation. The data element structure also permits a freedom in retrieving the information, which will give great usefulness in a wide variety of applications. Each data element is capable of functioning as an index point (and retrieval point) for the objective. Thus, one can search the file for specific data such as type of objective, subject area objectives, etc. The report generator, can print indexes using any data within any data element as an indexing heading.

The choice of the indexing model as a means of storing and retrieving was chosen after great deliberation. Logically, the problem involved two facets which had to be accommodated to successfully retrieve the desired set of statements. Uppermost in consideration was the problem of identifying statements concerned with problem solving, curriculum, specific courses, and task orientation. In addition, the idea of sequencing was considered significant. Since the competencies were created in a hierarchy, i.e., upper level objectives assume mastery of prior objectives, it was deemed essential that some means be provided to reflect this concept. Intelligent use of the file necessitates following a kind of path or track through the objective hierarchy.

Standard indexing techniques are used to provide solutions to the problems of identifying statements and proper sequencing according to the hierarchy. These techniques include identifying the terminology of the competencies, ordering the terminology into an hierarchy of broader and narrower terms, and describing each statement with appropriate terms from the hierarchy. In addition, classification descriptions will be assigned to identify the type of each statement (problem solving, curriculum, course or task oriented).

The third major application to be installed was an easy-to-learn authoring system. Several alternatives were initially considered and extensively evaluated. Among those considered were IDAF, Coursewriter, TICCIT, PLATO, PLANIT, and MULTITUTOR. The criteria for evaluting each system were that the language be flexible enough to be used with any instructional strategy, that it be powerful in terms of the number of functions which could be efficiently performed, and that it be easy to learn with little or no computer knowledge. Being essentially brash and immodest in our belief that our Network faculty could author their own courseware, the existance of courseware was not a primary criterion.

For a considerable period PLATO appeared to be the leading candidate for implementation on the Network system. For three reasons we ultimately selected MULTITUTOR. First, MULTITUTOR, of course, is very similar to PLATO Tutor which, in the opinions of those of us who know both systems, maintains the integrity of PLATO within a price range affordable to everyone. Secondly, MULTITUTOR combines some of the best features of PLATO with the ability to run on a wide range of terminals and at a nominal cost. The third reason for installing MULTITUTOR was the faculty time requirements. We have found that novices can learn to author simple lessons with approximately ten hours of instruction. Currently, lessons are being authored in numerous academic areas, and employ a variety of instructional strategies. There are about two hundred lessons now operational on MULTITUTOR at our site. We believe the work of Dr. Jim Schuyler in developing MULTITUTOR will have a great impact on the growth in popularity and acceptance of CBE in the future.

Other applications at various stages of development include a computer-assisted guidance system, an interactive testing system, a course evaluations analysis system, a student record keeping system, and an extensive interactive statistical analysis system. When these become commonplace to Network users other applications will probably come to the forefront as we continue movement toward a comprehensive set of CBE services.

Demonstrations and Training for Users

The existance of many applications available to users makes the need become apparent for demonstrating their value and teaching faculty members to use them effectively. Though this is true of all computing, it is especially important for CBE because of the impact CBE can have on teaching and learning and because most potential users know little about CBE.

In fact, the effort put forth in providing educational experiences for users is considerable and a realistic goal is the conversion of many of our present activities to computer-based educational strategies. Presently, we teach free, non-credit, courses and/or seminars each week for one hour every day. Topics for these courses include programming languages, job-control languages, computer center orientation, operating systems, utilities, and software packages. Most of the users who attend these sessions are graduate students.

CBE training is aimed at faculty members who want to author lessons and/or curricula for their own disciplines. The schedule for teaching about CBE is kept flexible to meet the needs of busy faculty members. Currently, we teach three types of sessions as follows:

1 - Orientation (1 hour)
    - CBE
    - Applications Available
    - Network Organization

2 - Using CBE (2 hours)
    - Orientation
    - Using the Central Site
    - Login Procedures
    - Accessing CBE Applications
    - Using CBE Applications
    - Hands on Experience

3 - CBE Skill Development (10 hours)
    - Orientation
    - Using CBE Applications
    - Building Test Item Data Bases
    - Instructional Design for CBE
    - CBE Authoring

Since September 1976, we have conducted these sessions an average of twice a month to groups as small as ten and as large as one hundred fifty. The response has been outstanding and the requests are still increasing within the Network.

Technical Support

Technical assistance is provided for planning, implementing, using, and evaluating hardware and software systems. The Division maintains files of descriptive material of user access equipment such as data terminals, printers, etc. for personal use by interested faculty. We also provide a central location for vendor demonstrations.

A staff of consultants is available during standard working hours to assist with all technical problems encountered by users. Users who want help with CBE usually come to the offices of the CBE specialists while other users can obtain aid at a user window.

The primary source of the technical assistance is provided to the Network by the circuit rider who spends much of his time around the State. He provides assistance with the remote job entry devices, the stand-alone systems in the Network, the interactive devices, and telecommunications.

In addition, the Central Site maintains a technical library of manuals, journals, locally written documentation, and video cassette training materials which can be borrowed by users.

ORGANIZATION

The Division of Educational Support functions within an administrative framework which was established at the time the network originated. The Georgia University System is headed by a Board of Regents appointed by the Governor. The Regents, in turn, appoint a Chancellor as chief administrative officer. The Presidents of each institution of higher education report to the Chancellor. The computer network is represented in the Chancellor's office by the Assistant Vice-Chancellor for Computing Activities. The specific organization of the Network includes local campus coordinators who function as liason between our Divisional central site staff and the users on each campus. These coordinators report to their respective presidents. The USCN Executive Committee, made up of six coordinators elected from each of six regions, act as a guiding body for reviewing and suggesting guidelines and policy for the Network.

The Division of Educational and Technical Support is housed in the Office of Computing Activities and consists of three departments:

1. The Department of Educational Support
2. The Department of Technical Support
3. The Department of Network Coordination

In actuality, a "project management" structure exists within the basic organization as outlined. The nature of many of the functions of the Division require that a combination of educational, systems design, and computer expertise be marshalled into a single project. This frequently results in formation of teams from the three Departments to work with a project leader. The project leader is usually the person with a background in educational applications.

PROBLEMS

The major problem in gaining state-wide acceptance of CBE in the Network is the relative lack of interactive terminals and related hardware at the network sites. This condition is rapidly changing but at present there are about 200 terminals distributed among 34 institutions - not an adequate number to provide continued CBE services at the network sites.

Another significant problem appears to be communicating the existence of ample services to the large number of faculty members throughout the University System. The problem of communications is being addressed through seminars, workshops and literature distributed throughout the network sites. A third apparent problem is the reluctance of data processing coordinators to enthusiastically endorse the concepts of CBE. This does not appear to be a significant problem where the responsible individual is an educator and has teaching duties; however, many of the coordinators are technical school graduates in Data Processing and do not recognize the importance of CBE in the teaching and learning process. In addition, there are more specific problems related to the utilization and development of CBE applications. Perhaps the most noteworthy is the problem of data entry which all authors eventually face. Faculty members can usually find time to perform the creative aspects of lesson design and development but are frequently inhibited by arduous tasks of entering the lessons into the computer. However, it is believed that MULTITUTOR is so easy to use that clerical personnel will ultimately be able to do lesson data entry for the busy faculty members. At present, the staff is planning an in-service program for secretarial personnel to train them in procedures for installing a lesson which has been developed.

Other specific problems involve locating good instructional materials, installing them on the system, and attempting to maintain some level of quality control over materials made public. These problems are currently under study in the hope that solutions will be found in order to assure future success of CBE in the University System Computer Network.

	Diagnosis of Learners Needs	Mgmt. of Inst. Objectives	Design of Tchng. Strategies	Implementation of Teaching Strategies	Coordination of Ed. Media	Monitoring Learner Achmnt.	Measuring Learner Achmnt.	Monitoring Instr. Effectiveness	Measuring Instr. Effectiveness	Generalized Management	Research and Development
CAI				X		X			X		X
CMI		X	X		X	X		X		X	X
CDI		X	X							X	X
CAT	X			X		X	X	X	X		X
CAG	X										X

FIGURE 1.

A NATIONAL SURVEY OF THE USES OF COMPUTERS FOR INSTRUCTION IN SCHOOLS FOR THE DEAF

James R. vonFeldt
National Technical Institute for the Deaf

Foreword

Educators of the deaf are becoming aware of the computer as an instructional tool. They are asking questions such as:

1. Is the computer being used for instruction of deaf?
2. Who is using it?
3. Is it effective?
4. How much does it cost?
5. How will it effect faculty?
6. How do I get a system to meet my school needs?

And many other questions. Unfortunately, there is little practical help available. True, there are research reports, evaluation reports and opinions (everyone has an opinion) but there is little comprehensive data to provide clear guidance. However, the following has been accomplished and more is on the way:

1. National study of CAI drill and practice with deaf. Dr. Patric Suppes et. al., 1969.
2. Development of CAI, drill and practice, tutorial and simulation for deaf students in post high school technical curricula. NTID, Dr. James vonFeldt et. al., 1968-1975.
3. Process for efficient CAI development. NTID, Dr. James vonFeldt, et. al., 1973-1975.
4. Implementation of numerous CAI systems for instruction of the deaf, 1970-1976.
5. Isolated evaluation and research reports specific to CAI units.
6. Specific research activities. Clarcq School, MSSD, Gallaudet, Ohio projects, NTID and others.
7. Formation of a national organization *ADCIS and a special interest group interested in education of deaf *SIGED through computer technology. SIGED formation, 1973, Ms. Linda Culbertson and Dr. James R. vonFeldt.
8. National survey of public secondary schools use of computers for instruction. American Institute of Research reports. Dr. Korotkin et. al., 1970 and 1975.

A significant problem exists in assembling appropriate information at an overview perspective.

Project Purpose

The purpose of this study is to establish a national base of data describing instructional computer activities at schools for the deaf. Special attention is paid to computer-assisted instruction (CAI).

---

* ADCIS, Association for the Development of Computer-Based Instructional Systems.

* SIGED, Special Interest Group For the Education of Deaf.

Project Design

In the spring of 1976 a questionnaire was designed and sent to three populations:

1. Schools for the deaf listed in the 1976 American Annals of the Deaf.
    a. All schools showing a student population of 100 students or more.
    b. In states that had less than 100 students, surveys were sent to the three largest schools.
2. Membership of the ADCIS special interest group for the education of deaf (SIGED).
3. Schools for the deaf that support computer science-data processing curricula.

All lists were compared to reduce redundancy. Two hundred and thirty-seven questionnaires (237) were sent out to 50 states. One hundred and eleven (111) were returned.

Questions

This survey addressed the following questions:

1. Are schools for the deaf using computer technology for instruction of their students?

If schools responding to the survey were using the computer, the following questions were asked:

2. How do you support your system (own, rent, timeshare, etc.).
3.
    a. What percent of your machine is available for instruction?
    b. Specify applications.
    (Two categories were provided: first, data processing and related; secondly, CAI and related).
4. Describe your machine/s and number of terminals.
5. Describe the age group of your student population.
6. Do you design and develop CAI?
    a. What language/s do you use?
    b. What training is available - describe?

7. If you are using CAI what is the main emphasis (remediation-classroom support, etc.)? Also: list currently used CAI courses.

8. Do you think CAI use will increase, decrease or stay the same in the next 5 years?

Questions 9 and 10 were filled out by both those who did have a computer system and those who did not.

9. What do you feel are the benefits of instructional uses of the computer?

10. What are the problems of instructional uses of the computer?

Survey Results

The results of the survey show a strong start toward the use of the computer and significant growth in the last 3 years. Eleven (11) of the fifty states surveyed have implemented computer based instruction. The survey identified 34 systems with a total of 408 terminals. A very small percent was allocated for computer science activity.

Most computer systems (16) are rented, turn-key systems that provide drill and practice in math and reading. However, surprising number of systems have been purchased (9). The remaining systems (9) utilize various time-sharing systems.

Mini, intermediate and large computer systems are represented. Specifically, there are specially designed instructional systems and adapted general purpose systems.

Of special note is the indication that the instructional systems are not generally shared with other applications. Twenty-eight (28) of the 34 systems are used strictly for computer-assisted instruction.

Mini-computers represent 20 of the 34 systems. One company, Computer Curriculum Corporation, dominates the current scene with 15 of their systems (173 terminals). The next nearest competition is IBM and DEC with 4 systems each and Univac with 3 (120 terminals). The fundamental difference between CCC's system and all the others is that it provides a turn-key system that provides ready-made math and reading curricula. It requires no technical expertise to maintain and has no capacity to develop new curricula.

For those developing their own materials, two statements can be made. First BASIC appears to be dominating language used (BASIC is the dominant language at public school installations). Secondly, instructional computer applications are occurring in a wide and diverse range of subject areas (also consistent with public school findings).

The survey shows that educators for the deaf are entering into the application of computers for instruction in a radically different way than was experienced by hearing public schools and higher education. The normal sequence of starting with research, then administrative, then computer science and finally to instructional applications (CAI) have been circumvented. Educators of the deaf appear to be going directly to CAI applications; 63 CAI applications reported: 18 computer science; 5 research; and, 7 administrative.

As would be expected, the populations served are elementary through high school but here again major differences occur as compared to public education across the nation. Schools for the deaf are serving the entire spectrum (elementary through high school) whereas public schools focus on elementary, junior high or high school with a specific system.

A surprising number of installations indicated that they were serving deaf adults (not enrolled in post high school). Few, but significant, computer applications exist at institutions for the deaf at post high school levels.

The respondee's outlook for the future was optimistic and extremely positive. They cited many experiences substantiating observed student as well as teacher benefits. Almost all (27) predicted growth of their systems in the next 5 years.

Problems that were cited most were the need for more terminals, capacity for course development, training for faculty, support (developmental) and access to courses and units of instruction developed at other institutions.

Conclusion

There has been a rapid growth in the implementation of computers for instructing deaf in the last 3 years. Observations by attending faculty and administrators are very positive about their experiences. However, no one system has proven that it can provide all of the needs for deaf students. Perhaps, at this early stage, all of the needs (appropriate applications) and the related computer capacities are not yet clearly known. One thing is clear though, this technology is having a strong effect in the learning environment.

Special Note

Numerous requests have inspired the author to efforts following the survey.

1. An Index of CAI courses and units of instruction used for education of deaf (due July 1, 1977).
2. Current listing of CAI installations at schools for the deaf.
3. A collection of CAI evaluations, reports, studies and research (due 1977).

To provide you with a meaningful service will require your help. If you have completed reports, installed systems or information relative to the above projects, please send it to:

James R. vonFeldt
National Technical Institute for the Deaf
Rochester Institute of Technology
Rochester, New York 14623

Phone: (716) 464-6370

The following information may also be obtained from Dr. vonFeldt:

1. Complete Survey Questionnaire.
2. List of schools for the deaf which support instructional computer applications.
3. Summary of comments on benefits and problems.

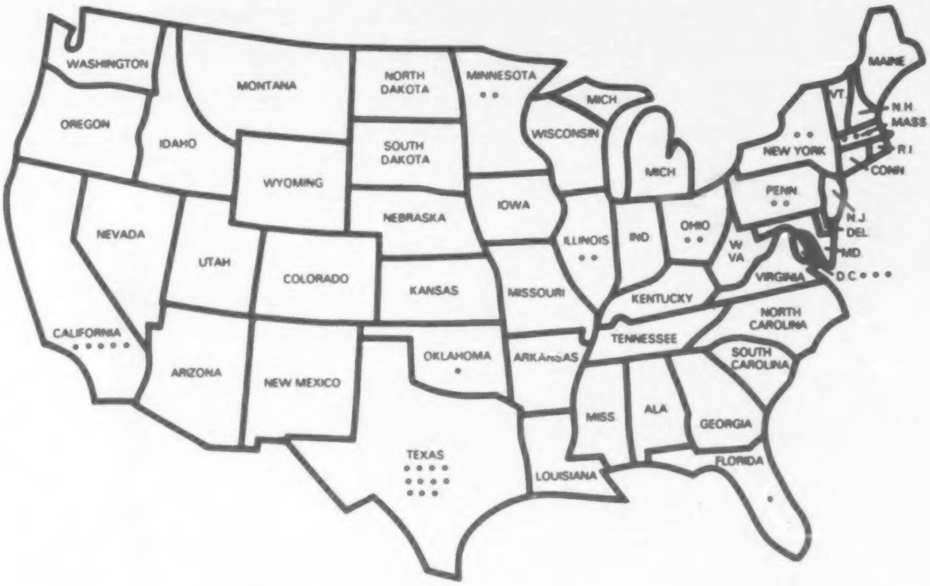
1. Do teachers have access to a computer at your institution?

237 Surveys sent

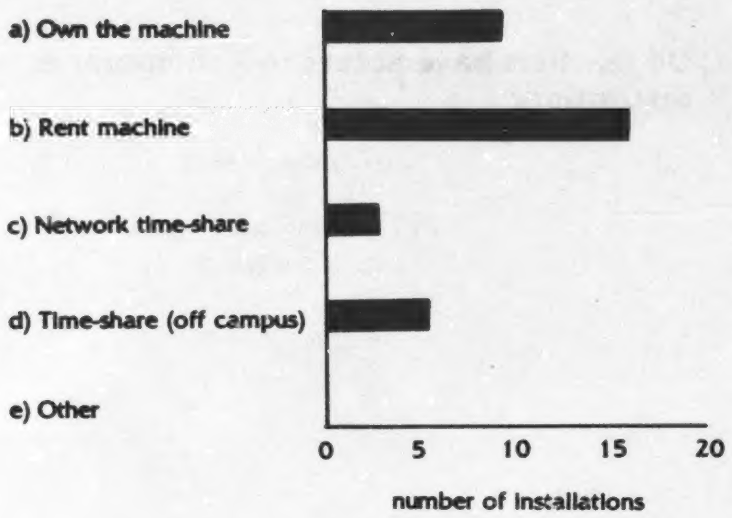
111 Surveys returned
(46% returned)

34-Yes 77-No

Locations of computer systems used for instruction of the deaf in the United States

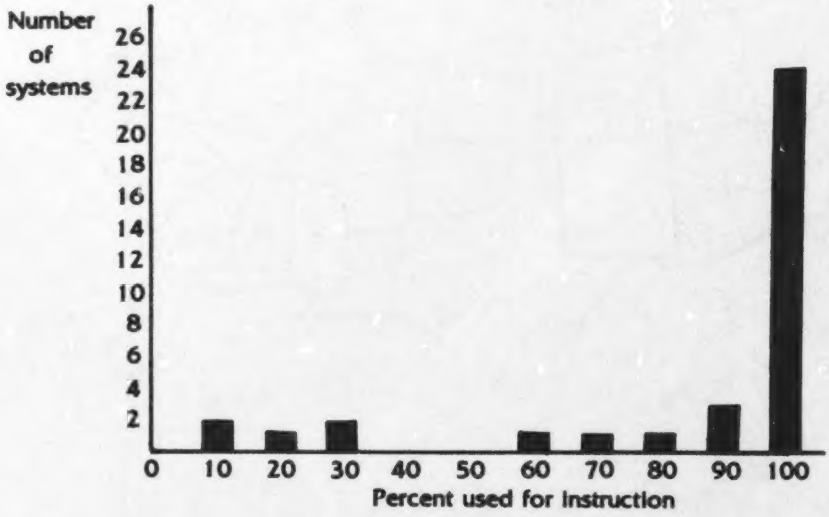

WASHINGTON
MONTANA
NORTH DAKOTA
MINNESOTA
MAINE
MICH
VT.
N.H
MASS
OREGON
IDAHO
SOUTH DAKOTA
WISCONSIN
NEW YORK
R.I
CONN
WYOMING
MICH
PENN
IOWA
NEBRASKA
N.J.
NEVADA
ILLINOIS
IND
OHIO
DEL
W VA
MD
UTAH
COLORADO
KANSAS
MISSOURI
VIRGINIA
D.C.
KENTUCKY
NORTH CAROLINA
CALIFORNIA
TENNESSEE
OKLAHOMA
ARKANSAS
SOUTH CAROLINA
ARIZONA
NEW MEXICO
MISS
ALA
GEORGIA
TEXAS
LOUISIANA
FLORIDA

2. If you do have access, in what ways is your computer/s supported?

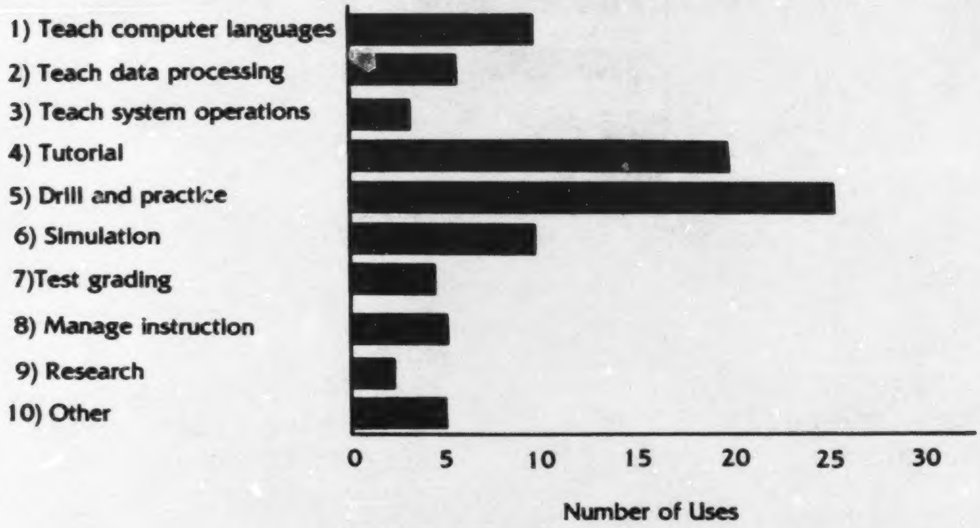


3. Estimate the allocation of your machine/s for instructional uses [in percentage of use].

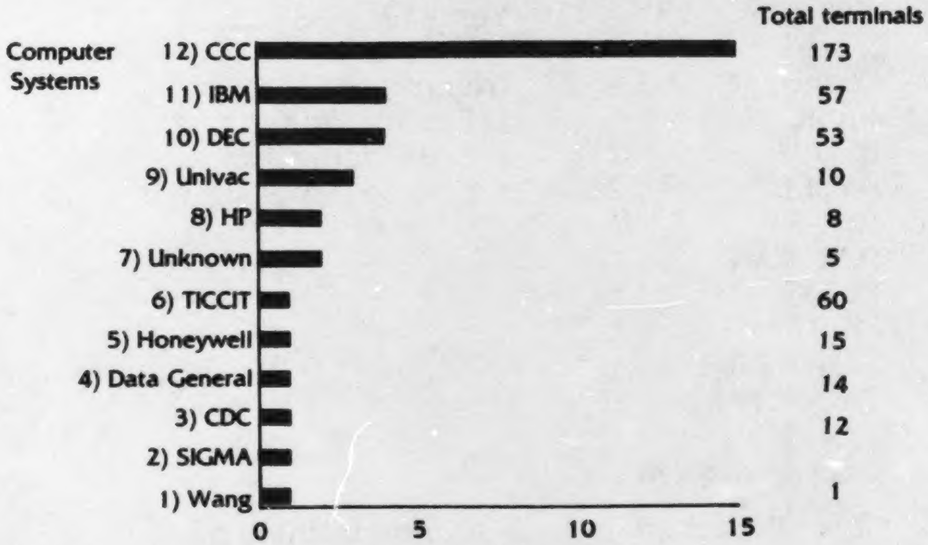
a. ___ percent of our machine use is for instructional purposes.



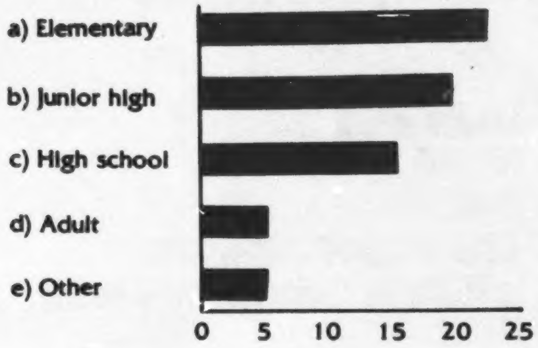
3. Estimate the allocation of your machine/s for instructional uses [in percentage of use].



4. Describe your machine/s [name, model, periferals number of terminals, etc.].



5. What is the age group of your student population?



Others:
College (2)
Tech-College (2)
Preschool (2)

Note:
29 of 34 responding marked multiple levels of students using the computer

6. Do faculty at your school design and develop CAI lessons?

No	22	67%
Yes	12	33%

Faculty Training?

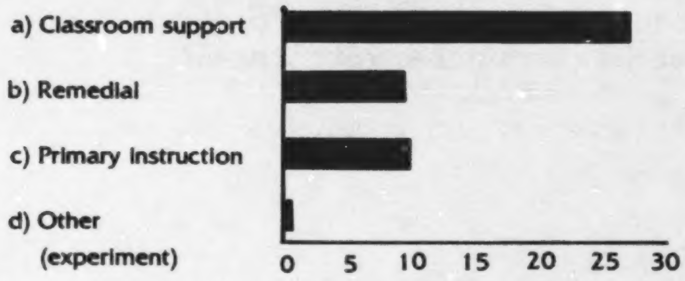
No	22	67%
Yes	12	33%

Language used	
BASIC	5
IDAF	2
ASCET	2
APL	1
FORTRAN	1
GNOS	1
PILOT	1
SCHOLAR-T	1
TACL-CW II	1
TICCIT	1
Coursewriter III	1

Training	
1. Proctor Training	2
2. CAI Orientation	7
3. Course Development	12

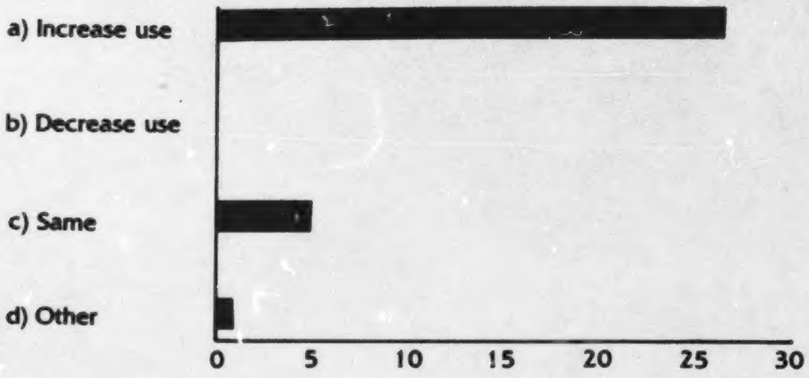
Note: Most installations offering training had orientation and course development training.

7. If you are using the computer as a CAI instrument - what is the main emphasis? (27 of 34 were)



Note:
There was a high incidence of multiple use reported

8. What do you think is going to happen in the next 5 years at your school; re, instructional uses of the computer?



9. What do you feel are the benefits of instructional uses of the computer at your school?

48 categories identified as benefits

Example:
"Provides Individualized Instruction"
"Provides High Motivation"
"Provides More Teacher Time for Working with Students"

10. What are the problems of instructional uses of the computer at your school?

21 categories identified as problems.

Example:
Teachers Needs:
a) to develop lessons
b) time for development
c) training for development

High cost of computers

COMPUTER-ASSISTED INSTRUCTION
AT
MINNESOTA SCHOOL FOR THE DEAF

Benet R. Freund, Project Director, Computer Services Coordinator,
Minnesota School For The Deaf, Faribault, MN 55021

The purpose of today's presentation is to share with the profession the goals and present status of the year and a half old computer-assisted instruction program at the Minnesota School for the Deaf in Faribault, Minnesota.

The Minnesota School for the Deaf program started as an idea brought up at a science curriculum meeting in January, 1975. The idea was discussed and one member of the science department decided to persue the idea because he was acquainted with some computer-assisted instruction work at the public schools in Faribault and it was thought that the Minnesota School for the Deaf could develop its own programs that could be designed especially for deaf students.

Funding for the first two years of the program has been supplied by the Minnesota Council on Quality Education, which is an agency of the Minnesota Department of Education that gives support to inovative and cost effective programs throughout the State of Minnesota. The contract with the Minnesota Council on Quality Education states that the Minnesota School for the Deaf will develop materials to be used in schools throughout Minnesota.

The present system has three terminals; one hard copy TTY; one thirty character/second, hard copy printer and a thirty character/second video terminal. All the terminals have been purchased and are under long tern service contracts. The computer is a Univac 1110, owned by the Minnesota Educational Computing Consortium (MECC). The Minnesota School for the Deaf has its ports acoustically coupled to it on a timesharing basis with about 300 other school districts throughout Minnesota. Response time and up time has been very good and response time usually been less than four seconds so student frustrations are very low in that area.

Last year was the start up year for the project and a total of thirty-two short programs were written by the director and two part-time programmers. Most of these programs were written for and have been developed in the area of physical science and they include drill and practise, problem solving, information retrieval and test and worksheet generation problems.

The programs developed at the Minnesota School for the Deaf have been aimed at a narrow population using only one physical science text, and thus lack the strand concept often found in other computer-assisted instruction programs that are designed to coordinate with multiple texts. It was felt that the only reasonable goal, given the time and monetary constraints on this project, was to develop a year of programs that would supplement the ninth grade physical science course as taught at the Minnesota School for the Deaf.

The goals of the project are to: 1. Develop CAI programs in science for deaf students, 2. Increase the depth of subject matter presented and, 3. To have the student's mean gain on the Stanford Acheivment Test equal

0.90 grade equivalent. The major evaluation of these goals and the effectiveness of the computer-assisted instruction program was derived from the science subtest scores on the student's Stanford Achievment Tests. The Stanford Achievement Test is given to all M.S.D. students yearly so that the difference between the test score on the science subtest at the end of the eighth grade and the science subtest given at the end of ninth grade has been used as the measuring instrument.

Thirty-four students were tested and eighteen (53%) of them gained more than one grade level with the range being from 0.0 grade level to 2.2 grade level with the mean gain 0.93 grade level. There are presently no built in evaluation mechanisms in the programs, but record keeping programs are being developed.

The gains made by the students have yet to be analysed over a period of years to see if the ninth grade classes of 1975-76 are a representative sample and the norms for the data are useable for ninth grade classes in future years. The project is evaluated yearly by an outside contractor and although all of the first year goals were not met, the gain of 0.93 grade equivalent on the SAT science subtest was a larger gain then was first expected.

FUTURE PLANS

Computer-Assisted Instruction at the Minnesota School for the Deaf presently depends on funding approval by the next session of the Minnesota legislature. The funding from the Minnesota Council on Quality Education has been $17,500 for the two years of the project and the outlook for future support looks good. The project will be expanding to cover parts of the meteorolgy, geology and chemistry courses in the coming year with plans to have mathematics use more of the project time as the physical science programs become more settled and the gaps in the programs are filled.

A COMPUTER-IMPLEMENTED MODELING AND PROBLEM-SOLVING APPROACH TO TEACHING STATISTICS

Stanley W. Steinkamp
University of Illinois at Urbana

Statistics courses too frequently turn students "off" rather than "on." Yet in our modern world, the techniques, tools, and concepts taught in these courses are essential for resolving many of the problems which students will later encounter in the real world. To turn students "on" we must not only teach techniques, tools, and concepts but also their practical relevance for resolving future problems. The computer-based approach presented in this paper focuses on teaching statistics in a way which demonstrates for students the relevance of statistics.

Traditional Instruction

Traditionally, classroom instruction in statistics has employed compartmentalization to obtain relatively simple, homogeneous areas which can be taught in a rigorous fashion. To give the appearance of realism to instruction, problems clothed in real world terminology, but which do not transcend the compartment being dealt with, are assigned and discussed. The difficulty with compartmentalization is that while it facilitates learning specific theory and technique, the student too frequently loses sight of the bigger picture and the relevance of statistics for resolving real world problems. Failure to see relevance turns statistics into an exercise in memorizing and manipulating irrelevant concepts and tools. It is no wonder students are not motivated to learn statistics.

Interactive Instruction

Interactive instruction based on evolving computer capabilities is being put forward as both a viable alternative and supplement to traditional classroom instruction. But a closer look at the method underlying interactive instruction reveals increased compartmentalization rather than less. A subject area is broken down into a number of homogeneous modules comprised of a set of structured questions for which computer responses can be identified and programmed. One would expect still lower levels of achievement and motivation with interactive instruction, but in fact a number of studies report higher levels of learning and motivation. Skavaril (1974) and Ozarowski (1973) both report achievement of higher levels of learning per student hour devoted to the course than was achieved in traditional sections acting as control groups. Other researchers using more casual evaluation methods report consistent results.

What then is the basis for the contradiction? A review of the work in the area beginning with Suppes and Morningstar (1969) through more recent work of Skavaril (1974), Ozarowski (1973), and Scalzo and Hughes (1976) identify imposed discipline, self-pacing aspects, clarity of presentation, frequent testing of comprehension as a prerequisite for proceeding, together with a basic enthusiasm, interest, involvement, and challenge associated with working on a computer, as the bases for improved performance.

The frequent reference to and continued emphasis on the challenge of working with the computer raises the question of whether it is the media or the message that is motivating students. A number of students who were taking their second, third, and in one case a fourth interactive course at the University of Illinois at Urbana reported, in casual conversation, that although they still liked interactive instruction—the self-paced aspect, the imposed discipline, the review and convenience--they felt that the stimulation and challenge of the computer and of interactive instruction were definitely declining for them. The implications as universities shift to more widespread use of interactive instruction is clear. A rigorous study of the time and breadth of use effects is needed.

Critical-thinking, Problem-solving Approach

One enduring source of motivation is student realization that what is being taught is relevant for achieving their future goals. One way to accomplish this is to present the student with ambiguous, complex, real-world problems and then develop the necessary set of concepts, tools, and techniques to resolve the problems. In an attempt to make statistics relevant, Brightman and Broida (1975) developed a course around this orientation to be used in the second semester of a two-semester sequence in Business at Miami University. Employing a critical-thinking, problem-solving approach based on suggestions by Hogg (1972) they report that:

> Defining real-world problems forces the student to analyze the problem situation, separating symptoms from fundamental causes. Solving real-world problems forces the student to synthesize the available subjective and objective data.

The emphasis in their course went beyond the analysis of real world data bases as a means of increasing student motivation. Rather they emphasized diagnosing the problem situation to determine what data were necessary. Statistical verification techniques were only introduced in the final step of the problem resolution process.

The problem-solving methodology employed was based on Dewey's (1933) five stages of critical thinking. The uninstructured character of the approach is indicated by the topics included in the course: 1) problem identification which focuses on identifying deviations from desired norms; 2) a suggestion phase which involves generating possible solutions through brainstorming and synetics; 3) hypothesis formation; 4) reasoning where the logical consequences of a set of hypotheses is derived; 5) a verification phase where the canons of agreement, differences and concomitance form a frame for teaching hypothesis testing, analyses of variance, and regression analysis.

In implementing this approach, complex real world types of problems presented in case form were employed. One such application involved a problem in hardening frames in the production of eye glasses. A second involved the intermittent failure of a sidewinder missile.

When compared with a control section in the course who received the traditional approach, Brightman and Broida found that students receiving the critical-thinking, problem-solving orientation did significantly better on a comprehensive test covering ANOVA and Regression even though these subjects were taught in the same way and by the same instructor as was used with the control section. Brightman and Broida attribute this to a significantly larger number of students in the critical-thinking, problem-solving section as compared with the control section who felt that what they had learned was relevant for the problems they face or expected to face.

While this approach makes statistics more relevant and appears to increase motivation, it also severely limits the amount of traditional theory and technique taught. Whether this is good or bad depends on whether a student with limited traditional statistics but who understands how to apply what he has learned to solving ambiguous, complex problems will be more "effective" than one with a thorough dose of traditional theory but who is uncertain about how to use this knowledge.

Computer-implemented Modeling and Problem-solving Approach

What is needed is a synthesis of the three approaches which would motivate students through making statistics relevant and would teach the theory and skills necessary for applied work. Given the usual one- or two-semester time constraint, the synthesis must involve more than just a simple combining of the materials. The purpose of this paper is to describe a synthesis of the traditional, interactive, and problem-solving approaches--a synthesis to be called a Computer-implemented Modeling and Problem-solving (CIMP) approach--and give a preliminary report on the success which has been achieved with this kind of instruction. In synthesizing the three approaches it is necessary to orchestrate the numerous parts and pieces into a new approach. Several motifs dominate.

First, the basic problem-solving orientation of Brightman and Broida is retained. Within the first week of class students are given a broadly defined problem for which they must learn or develop global concepts and tools to resolve it. However, rather than tackle actual ambiguous and complex problems, the student is given an extensive write-up of a business and its markets which is simulated in the computer. Students can input decisions on controlled variables and can observe the resulting impacts on the business situation. They are instructed to design a multistage approach to identifying and resolving the problem. Each student has a unique simulation so that students can consult with each other about method but not on final answers.

A critical element in using a broad problem-solving orientation is to present students with problem situations which have an acceptable degree of realism but where the necessary theory and technique required to solve the problem can be taught without the student losing sight of the overall problem. A program SYSGEN developed as part of the CIMP approach not only generates unique business simulations for each student, but can also be programmed to allow the instructor to match the difficulty of the problems to evolving student abilities. This is particularly important early in the semester when, in the interest of motivation, students should experience the resolution of a complete global type problem.

Concurrent with this assignment, an overview to problem resolution is discussed in class. Among the areas covered are: 1) the identification of the problem and formulation of objectives, 2) identification of alternatives, 3) design of experiments to learn about properties of the system, 4) selection of a model system and establishing correspondences between the parts of the system and symbols in the model system, 5) estimation of parameters of the model, 6) evaluation of the model's predictive and explanatory value using independent data from the business system, and 7) massaging of the model to identify the best solution to the original problem.

This overview serves two functions. First, in applying it to the assigned problem, students identify needed concepts and techniques necessary to complete the assignment. These concepts and techniques frequently turn out to be traditional theory and technique which can be discussed within this global context. Second, given this initial overview, the remainder of the semester is devoted to elaborating on and developing increasingly more sophisticated problem-solving concepts and theory.

Solution-generating techniques such as brainstorming and synetics taught by Brightman and Broida are not covered in the CIMP approach. It is felt that if a course in synthesis is to be offered it must follow and build upon a course such as the one described here which emphasizes analysis.

A second motif is built around instruction modules covering theory and technique identified or required at the global level of problem resolution. These modules are lifted out of the traditional approach and tie the traditional text used in the course into the CIMP approach. This material is partially supplemented with interactive lessons available on the University of Illinois' DEC system 10.

The third motif, interwoven in the counterplay between the global problem-solving level and the specific theory and technique level, is discussion and development of models and of modeling techniques. A survey of the leading statistics texts revealed that "models" and "modeling" are not discussed in the introductory chapters and received infrequent explicit reference in subsequent chapters. This stands in stark contrast with the leading econometric and operations research texts where models and modeling are extensively discussed and explicitly used throughout the text. When one considers that econometrics and operations research are the applied frontiers of statistics, this fact seems appalling.

In discussing models, the point is made that a model system is one in which the relation between the symbols or parts of the model system are the same as the relation between the parts of the real system for which they stand. When this condition is achieved, then by manipulating the model system one can draw conclusions about the behavior of the real system. This separation of the model system from the system to be modeled is maintained throughout the course. It is easy then to talk about the properties of a potential model system and whether, given a set of correspondence rules, these are similar to those observed in the system to be modeled. This type of an introduction leads naturally into problems of building models and of drawing inferences from them.

A final motif in the CIMP approach is the repeated use of traditional theory and technique within a problem-solving frame. Past experience with students indicates that repeated application of traditional theory to problem solving is a critical factor in their coming to perceive statistics as relevant to personal long-term goals. Ten problem-solving assignments are made during the semester with the difficulty and complexity increasing through the semester. Rather than spend more time developing the CIMP approach, specific cases of its implementation will be discussed.

Implementation of CIMP Approach

Students are initially assigned a problem involving a firm with a limited cash flow and three areas which might increase worker productivity and the firm's profits. The students are to 1) identify the problem and define objectives, 2) design experiments to learn how the areas are related to profits (what areas have an effect and do areas interact in their effect on profits), 3) select an appropriate model system from their bag of model systems (it has a linear and power function at this point), 4) estimate parameters of the model, 5) evaluate accuracy of the model in terms of predicting the firm's behavior, and 6) use the model as a basis for identifying the best action for the firm. In this first problem the situation being simulated is deterministic.

Students could experiment with and obtain data from the simulation of the firm using the batch facilities and running under EXPRESS (one minute average turnaround time). They were given sufficient funds for three runs; with complete planning and design of experiments one run would be sufficient, although most used two runs--the first for identifying properties of the system and building the model and the second for evaluating it. Since students make errors, successful completion of the problem in three runs requires prior overall planning of a multistage approach and identification of the data they would require to complete the problem.

The reader may be asking at this point why a deterministic system is studied in a course dealing with stochastic systems. There are a number of reasons.

1. The situation is simple enough that a student can be given a complete problem-solving assignment which, except for being deterministic, is realistic. It was found that 94% of the students in the CIMP section completed the problem and wrote it up within four days after it was assigned. Further, students were enthusiastic about building their "first model" which predicts the behavior of a business system.

2. It forcefully drives home the distinction between the model system and the real system. The student gets the notion that he must set up correspondence rules between symbols in the model system and parts or elements in the actual system. Then he must set about to build into the model system the same relation between the symbols as exists between the parts of the system to which the symbols correspond.

3. A third reason relates to goals for the course:

   a) The regression model is composed of two sub-models.

   $$y = \mu_{yx} + \xi \qquad (1)$$

   where

   $$\mu_{yx} = B_0 + B_1X + B_2X_2 \cdots \qquad (2)$$

   $$\xi \text{ has some distribution} \qquad (3)$$

   but note that Equation (2) is a deterministic relation.

   b) It is very natural to talk about specification error, particularly when students must select between a linear or power function in modeling the system.

4. Students come to really understand what a parameter is. The analogy is useful when talking about the parameter of a distribution.

5. They can more readily grasp simulation which is used later in the course to study the population sample inference problem.

For the next problem assigned in the CIMP approach, the simulated business studied in the first problem has a random element added. Students are instructed to submit 30 decisions where expenditure in each area is held constant at the optimal level. Given a listing of the 30 resulting profit levels, they are told to build a model of the observed variation in profit. In spite of the fact that they have all had a first semester course which covered distribution models, most are at a loss on what to do. However, when instructed to prepare a histogram or frequency polygon, the light dawns. Having estimated parameters for the distribution they must draw inferences about profits at the optimum level.

Two-sample hypothesis testing follows directly using a new scenario; they now must identify which of three variables affect sales. As part of this problem they must intentionally generate a type two error. This they can do by changing a significant variable by a small amount so that $\mu_1 - \mu_2$ is small. Now the true population distribution lies largely in the acceptance region, yet $\mu_1 \neq \mu_2$.

After completing a number of others problems involving chi square, analysis of variance, and simple regression, they are given a scenario which involves multiple regression. They have now acquired a set of tools so that they can construct a model of a stochastic system similar to the one they built of a deterministic system the first week of the semester. However, we now employ a standard statistical package to do the computations and focus on evaluating assumptions and specifying the model. The package currently employed is SOUPAC (1976). This package has a number of features not available in other packages.

The alternative to use of standard packages is the use of special programs such as Skavaril and Scalzo and Hughes reported using. The use of special programs is rejected for three reasons. First, when students accept jobs at universities and businesses, the special programs will not be available. Standard statistical packages, if not available, can be obtained for a nominal yearly fee. Second, the reliability of special programs, particularly when used with ill conditioned data, is likely to be questionable. Third, the standard packages generalize to computation in a very large number of situations; although the package is used in a specific regression context, students come to recognize its potential broad application.

Since SYSGEN, the program which generates unique problems for each student, can be set to generate specification errors, autocorrelation, and heteroscedasticity, the students can become familiar with identification of these problems. Multicolinearity can be intentionally introduced by students as can specification errors.

The final assignment for the students is to build a model of any system they wish. One student built a model to help his father decide where to locate filling stations in the Chicago area. Another built a model to project profitability of taking on a Coors distributorship. A student interested in mechanics built a model to explain sports car economy and acceleration characteristics. Still another student modeled a dating service in order to predict who users of the service are likely to be, so he could select advertising media on campus.

Evaluation of CIMP Approach

The CIMP approach has been continuously employed at the University of Illinois in one section of Economics 173--Statistics II since the fall semester of 1975. Since data for the 1976 fall semester are not yet available, the results reported here cover the first and second semesters of the 1975-76 academic year.

Although most of the 10 to 15 sections of E173 offered at the University of Illinois are staffed by teaching assistants, several senior faculty generally teach a section. The author taught one section during each of the semesters and sections taught by other senior faculty served as control groups. Since the fall of 1975 was the first time the author taught E173 and since he was not listed in the timetable as teaching the course in the spring of 1976, bias resulting from students selecting sections on the basis of instructor did not occur.

The CIMP section was compared with the control sections in two ways. First, the standard class evaluation questionnaire (CEQ), which is available to all students for evaluation of courses taken at the University of Illinois, was completed by all students in both the CIMP and control sections. Due to a special study of the Economics Department, however, different forms of the CEQ were used in the fall and in the spring. Second, student achievement was evaluated by administering to the CIMP section selected questions from the comprehensive final examination prepared for the regular sections. The selected questions did not include items on decision theory (because it was not covered in the traditional manner in the CIMP section), but did include all questions dealing with two-sample hypothesis testing, chi square, regression, time series, and indexes.

Results from the fall semester are presented in Table 1. Students were asked to respond to CEQ items 1 through 3 on a 6 point scale ranging from excellent to very poor. Items 4 through 6 were to be assigned a grade A through E. Students in the CIMP section, when compared with those in the control sections, gave significantly higher ($p < .01$) ratings in response to all six items. Of particular interest are the groups' responses to statements 2, 4, and 5. CIMP students were significantly more enthusiastic about the method of instruction, felt more strongly that "what learned is consistent with long range educational goals" and were more positive about the course in general.

When compared in terms of answers to selected questions on the comprehensive final examination, students in the CIMP section scored higher on the average than did the control sections, but the difference was not significant. What is not reflected in this result is the additional knowledge CIMP sections achieved relating to problem solving.

In the spring semester, a different and longer CEQ was used. It contained 23 statements which students were to scale on a five-point scale ranging from "strongly agree" to "strongly disagree." Responses to 9 of the 23 items are presented in Table 2, along with two composite scales based on the remaining 14 items. These evaluated the instructor, specifically and in general.

The statements can be grouped to reflect three areas of evaluation. Responses to statements 1, 3, and 8 reflect student feelings about the method of instruction. On all three items the CIMP approach yielded significantly ($p < .05$) higher ratings. Items 6, 9, and 20 reflect attitudes about course content. Again, all ratings were significantly ($p < .01$) higher for the CIMP section. Finally, statements 13, 16, and 23 relate to the course in general. Here again the CIMP approach is rated significantly higher ($p < .01$). These results are reinforced by the numerous comments "written-in" by students on the CEQs. Frequently the students commented that the course gave them something practical which they expected to use later.

Table 1

Fall Semester 1975-76

CEQ Items	Means CIMP Section	Means Control Sections	t ratios (df = 66)
1. Course content	4.50	3.56	3.17*
2. Method of instruction	4.83	3.09	4.29†
3. Major instructor	5.08	3.25	4.54†
4. What learned is consistent with long range educational goals	3.91	2.38	3.74†
5. Developed increased appreciation for course	4.08	2.59	3.63†
6. Course in general	4.33	2.94	4.54†
	N = 12	N = 56	

Selected Questions from Comprehensive Final Exam

	Mean CIMP Section	Mean Control Sections	t-ratio (df = 72)
Possible 57 points	36.9	35.7	.57
	N = 13	N = 61	

---

* difference significant at .01 level

† difference significant at .001 level

Might these differences be attributed to a superior instructor in the CIMP section? Composite ratings at the bottom of Table 2 show that this is not the case. Under "Instructor specific," the composite rating is based on responses to six statements relating to: personal interest in students, synthesis of course material, integration and summarization, knowledge of subject matter, viewing teaching as a chore, and development of new viewpoints. There was no significant difference between sections on these variables. The statements "instructor was excellent" and "would take another course from this instructor" formed the basis for the "Instructor-general" composite rating. Here too the difference between sections is not significant. On the basis of these results, the hypothesis that the quality of the CIMP instructor explains the observed differences must be rejected. The fact that the higher ratings given the CIMP approach were not a function of "instructor" indicates that the CIMP approach can be effectively used by instructors of varying personalities and abilities.

Table 2

Spring Semester 1975-76

CEQ Items	High Scale	Means CIMP Section	Means Control Sections	t-ratios (df = 83)
1. I would take another course that was taught this way	SA	3.64	2.77	3.21†
3. I would have preferred another method of teaching this course	SD	3.27	2.52	3.05†
6. Not much was gained by taking this course	SD	3.64	2.93	2.96†
8. I learn more when other teaching methods are used	SD	3.09	2.55	2.17*
9. The course material seemed worthwhile	SA	3.55	2.90	3.10†
13. It was a worthwhile course	SA	3.36	2.68	2.87†
16. This was one of my poorest courses	SD	3.50	2.91	2.37†
20. The course content was excellent	SA	3.18	2.50	3.02†
23. Overall, the course was good	SA	3.45	2.96	2.33†
Composite Ratings				
Instructor—General		3.41	3.05	1.38
Instructor—Specific		3.44	3.41	.30
		N = 11	N = 74	

Selected Questions from Comprehensive Final Exam

	Mean CIMP Section	Mean Control Sections	t-ratio (df = 90)
Possible 140 points	86.3	85.4	.91
	N = 14	N = 78	

---

* difference significant at .05 level
† difference significant at .01 level

At the bottom of Table 2 the mean scores of the CIMP and control sections on selected questions from the comprehensive final examinations are presented. Again, the CIMP section achieved higher mean scores, but they were not significantly higher.

Class size as a possible explanation for the observed differences has not been explored. In the fall semester the CIMP section was small because it was offered at eight o'clock when enrollment is traditionally small. In the spring semester, the author was not scheduled to teach a section of E173 as part of his normal duties, but decided to take a section on an overload basis. A regular section was split in order to make this possible. The impact of class size can be examined more rigorously when the results are available from the fall 1976 semester when there were 46 students in the CIMP section.

In summary, it appears that the CIMP approach to teaching statistics does in fact make statistics relevant for students. This is an important step because relevance as a motivating force will not be affected by time or the breadth of its use.

References

Brightman, H. and Broida, M. "On Problem Solving, Motivation and Statistics," The American Statistician, 1975, 29, 164-166.

Dewey, J. How We Think, New York: D. C. Heath, 1933.

Hogg, R. V., "On Statistical Education," The American Statistician, 1972, 26, 8-11.

Ozarowski, P. C. A Study in the Design and Implementation of a Course in Basic Fundamentals of Statistics via a Computer. Unpublished Doctoral Dissertation, University of Alabama, Montgomery, Alabama, 1973.

Scalzo, F. and Hughes, R. "Integration of Prepackaged Computer Programs into an Undergraduate Introductory Statistics Course," Journal of Computer-Based Instruction, 1976, 2, 73-79.

Skavaril, R. V. "Computer-Based Instruction of Introductory Statistics," Journal of Computer-Based Instruction, 1974, 1, 32-40.

SOUPAC Program Descriptions. CSO Vol. 9, Nov. 1975, Computing Services Office, University of Illinois at Urbana, Urbana, Illinois.

Suppes, P. and Morningstar, M. "Computer-assisted Instruction." Science, 1969, 166, 343-350.

Learner Control and CAI: Assumptions, Findings, and Questions[1]

Esther R. Steinberg

University of Illinois at Urbana-Champaign

CAI brings to education a whole new dimension and consequently a whole new set of questions that need to be addressed. One such is the question of whether CAI should be used as just another educational alternative or whether it should be used with the hope of improving instruction in some systematic way. The majority of experts polled by Anastasio (1974) agreed that the emphasis on computer use should be in developing unique capabilities such as the capacity for individualization. Although some technological approaches to individualization such as programmed instruction already exist, CAI offers a much greater multiplicity of dimensions (Bunderson, 1970). In particular, learner control of instruction seems to be well suited to the adaptive potential of CAI.

Assumptions

What is learner control, or what should it be? The opinions of the group of experts cited earlier were summarized by Anastasio (1974) this way: (1) the computer should be used as an intellectual resource, controlled by the student; (2) students should be taught appropriate strategies for obtaining and using information; and (3) learner control is the most direct approach to individualization but is effective only to the extent that the student can judge what works best for him. Thus some of the experts acknowledged that some students may not be competent judges of suitable learning strategies but other experts attached no such qualifications to their conception of learner control. As will become apparent from the review of empirical evidence, the latter assumed that either the student knows how or will somehow learn how to manage his own instruction. Other assumptions implicit in some of the above opinions are: (1) if the student does know the best or most suitable learning strategy, he will use it; (2) students really care or want to control their own instruction; and (3) just because it is available, learner control is necessarily good for all students.

Empirical Findings

What empirical evidence is there to support these assumptions? Under what conditions has it been demonstrated that the student knows how to learn? Consider first the cases in which students had essentially complete control. In a remedial college mathematics program, Judd, Bunderson, and Bessent (1970) arranged one learner control condition which included choice of a sequence of topics, amount of practice, and the option to leave a topic once it had been entered. Performance was worst for students under learner control, particularly for those who had done poorly on the pretest. In cases where students were given control over the course flow, as in the TICCIT English and mathematics projects (Bunderson, 1976) and in a Russian course (Curtin et al., 1976) on PLATO, they turned out to be poor managers of time. That is, many students did not complete the course within one semester, and some did not complete it at all. Achievement of TICCIT students in mathematics was apparently not lower than regular classroom students. Those students who did not do well showed inefficient learning strategies.

---

[1] This work was supported in part under contract US Army/DAHC 15-73-C-0077. The author wishes to thank R. A. Avner, L. Francis, A. Himwich, and R. G. Montanelli, Jr. for their critical reading of an earlier draft.

When a single variable was under student control, results varied with condition. Seidel (1975) found that even when students were chosen on the basis of aptitude for learning computer programming, not all of them used good learning strategies. But high performers were best in effectively choosing lesson sequences. Fry (1972) had students learn about computers by allowing some of them to decide which 4 of a set of 52 questions to ask to have answered. Yoked students saw the same 4 questions in different orders. High inquiry, high aptitude students did well under learner control, but high aptitude, low inquiry students did better under low student control. This handful of studies can not possibly serve as a basis for making sweeping generalizations. But it can serve as an indicator for formulating tentative hypotheses. There seems to be a consistent relationship between student performance and efficient learning strategies. Poor performers apparently do not choose efficient learning strategies. Good performers are more effective in choosing appropriate lesson sequences.

With respect to difficulty level, there is evidence so far that students are not particularly good at choosing it. At the elementary school level, for example, children in fourth through sixth grades were given an arithmatic drill and allowed to choose the difficulty level of the problems. They seemed to fall into two groups. Some chose problems that were too hard, while others chose problems that were too easy (Fisher et al., 1975). At the college level, Whitlock (1976) developed a generative exam system for a beginning computer science course. He allowed students to choose the difficulty level of their questions. They picked problems that were too hard. This may have been due in part to the fact that they were told they would get more points for the harder problems. Results are equivocal with respect to students' judgment of the amount of practice they need. Judd and his colleagues (1970) found that students were good judges, but Lahey, Crawford, and Hurlock (1975), in a military setting found "students overlearned." Atkinson (1972) set up a paired associate task for learning a foreign language vocabulary under three conditions: random presentation, student controlled, and response-sensitive computer controlled. During learning the greatest gains were made in the random condition, followed by learner control, and then computer control. But on a retention test performance was effective in reverse order. His goal was optimization of learning. For his purposes, he concluded that the computer is a better decision maker than the student.

There is no study thus far that has attempted to demonstrate that students learned how to learn by virtue of controlling their own instruction. In fact, Bunderson (1976) has suggested the need for some sort of "advisor" less primitive than the one built into the current TICCIT programs.

If the student does know an appropriate learning strategy, or if an advisor suggests one, will the student follow it? The answer is yes, but again only under specific conditions. In some computer science Fortran format drills, Montanelli and Steinberg (1976) either required or advised students about a suitable amount of practice. The students under learner control followed the advice as long as the exercises were not too difficult. That is, when the average percent of exercises done correctly in a set was about 75%. However, in the situation where the problems were more difficult and the initial error rate was high, students left before completing the suggested amount of practice.

What about motivation? Is learner control good because it maintains task engagement or creates a positive attitude? The results to date are equivocal. In some studies it does not affect attitude (Judd, Bunderson, & Bessent, 1970; Denton & Woods, 1975); in others it is very positively affected (Fry, 1972; Fisher et al., 1975). The problem is that a positive attitude is not always accompanied by good performance. In both the Fry and Fisher studies performance was worse.

Do the students care about having control over their own instruction? Again the answer is not clear. At the college level, for example, the answer may depend heavily on whether the course is required, something to be tolerated until completed, or an interesting elective. In the case of a computer science course required for commerce majors, the students indicated they really didn't care about the locus of control in a CAI lesson in Fortran arithmetic (Steinberg & Montanelli, 1975). When 16 alternative strategies were available to military students in a training lesson, for all practical purposes they chose to use only three different ones (Lahey & Crawford, 1976). Moreover, although they had the option of changing difficulty levels, they made little use of it.

Questions

The evidence thus far indicates that there is still a lot to be learned about learner control. It is clear that it is not universally successful, despite deep convictions that it ought to be. On the one hand, nothing is more frustrating to a student than being so controlled by a CAI lesson as to be unable to continue with the lesson. For example, suppose that in a drill, the student must do six consecutive exercises correctly. If after doing five correctly, he makes a typing error on the sixth, the computer requires him to do another six before he is allowed to continue. This is really abrasive. On the other hand, it is obvious that not all students are capable of making appropriate educational decisions. Even a generally bright student may not have suitable learning strategies for material for which he has no previous experience, such as a social science major studying computer science.

How does one decide what choices to leave up to the student? What is the precise role of learner control in CAI? Why should such an option be available? Under what conditions is it motivating, or at least enough of an attention maintainer to prevent students from dropping out? Can the student become a better learner if some aspects of the learning situation are under his control, and guidance is provided to help him learn how to make reasonable decisions? Is it realistic to expect to have much impact when considered in the total educational environment? That is, CAI is such a small part of education. How effective can it be in teaching learner control when so much of education is highly structured?

Summary

The idea of individualizing instruction by allowing learner control seems well suited to CAI. The problem is that what is good for some students under some circumstances is not universally good. There is not much point in providing so much control that motivation is maintained at the cost of a decrement in learning. But we have yet to learn how to maintain a balance between the student and the computer as the locus of control.

References

Anastasio, E. J. Computer-based education: Obstacles to its use and plans for future action. *Viewpoints*, 1974, *50*, 11-37.

Atkinson, R. C. Ingredients for a theory of instruction. *American Psychologist*, 1972, *27*, 921-931.

Bunderson, C. V. The computer and instructional design. In W. H. Holtzman (Ed.), *Computer-assisted instruction, testing, and guidance*. New York: Harper & Row, 1970.

Bunderson, C. V. TICCIT courseware development report. Paper presented at the meeting of the American Educational Research Association, San Francisco, April 1976.

Curtin, C., Dawson, C. L., Provenzano, N. & Cooper, P. The PLATO system: Using the computer to teach Russian. *Slavic and East European Journal*, 1976, *20*.

Denton, J. J. & Woods, B. G. A computer-managed instructional program in high school physics. *Southern Journal of Educational Research*, 1975, *9*, 188-202.

Fisher, M. D., Blackwell, L. R., Garcia, A. B. & Greene, J. C. Effects of student control and choice on engagement in a CAI arithmetic task in a low-income school. *Journal of Educational Psychology*, 1975, *67*, 776-783.

Fry, J. P. Interactive relationship between inquisitiveness and student control of instruction. *Journal of Educational Psychology*, 1972, *63*, 459-465.

Judd, W. A., Bunderson, C. V. & Bessent, E. W. *An investigation of the effects of learner control in computer-assisted instruction prerequisite mathematics* (MATHS Tech. Rep. 5). Austin, Texas: University of Texas. (ERIC Document Reproduction Service No. ED 053 532).

Lahey, G. F. & Crawford, A. M. Learner control of lesson strategy: Some tentative results. Paper presented at the meeting of the American Educational Research Association, San Francisco, April 1976.

Lahey, G. F., Crawford, A. M. & Hurlock, R. E. Use of an interactive general-purpose computer terminal to simulate training equipment operation. San Diego, Ca.: Navy Personnel Research and Development Center, Report TR 76-19, November 1975.

Montanelli, R. G., Jr. & Steinberg, E. R. Uses of ACSES in instruction. In J. Nievergelt (Ed.), *ACSES: The automated computer science education system at the University of Illinois (Report no. 810)*. Urbana, Ill.: University of Illinois, Department of Computer Science, 1976.

Seidel, R. J. Learner control of instructional sequencing within an adaptive tutorial CAI environment (HumRRO Tech. Rep. 75-7). Alexandria, Va.: Human Resources Research Organization, June 1975.

Steinberg, E. R. & Montanelli, R. G., Jr. Aspects of human-machine interaction in a CAI lesson. Manuscript submitted for publication, 1976.

Whitlock, L. R. Interactive test construction and administration in the generative exam system. Unpublished doctoral dissertation, University of Illinois at Urbana-Champaign, 1976.

DATA ANALYSIS and RETENTION: EXPERIENCE and RECOMMENDATIONS

Paul Lorton, Jr. and Rosemary N. Killam
University of San Francisco University of Delaware

I. INTRODUCTION

How do we know when Computer-Assisted Instruction (CAI) is effective? Our usual pre-test, post-test methods give us very global comparisons of CAI with Teacher-Assisted Instruction (TAI?) or some other variously assisted instruction. Another popular method of testing effectiveness is systematic hand waving - one looks at the student before and after CAI and, as long as the student wasn't noticably damaged, attributes resulting changes to the computerized instruction.

Usual rigor means pre- and post- treatment data are collected and compared to make an essentially binary decision on whether the instructional treatment was effective or produced "different results." While these comparisons are important, they do not advance the state of the art as systematically as more detailed studies of teaching/ learning by computer - studies which analyze response by response progress of the student. This second type of study has also been widely accomplished with results which tend to refine our art and produce knowledge basic to understanding the instructional process itself.

Replication of either type of study is important. With results reproduced from time to time and system to system, a body of findings develops which forms the solid foundation for building effective systems for instruction. With the first type of study, replication is a straightforward job of using similar pre- and post- data collection devices to compare a similar collection of variously assisted instruction. With the latter, more intensive study replication and direct comparison are more difficult. We feel that this difficulty is not inherent in the method nor in the sophistication of the available techniques but in the will of those doing the study to plan for such replication.

The use of computers for instruction is being done on an increasing variety of systems offering a great range of student-system interaction. It is of great interest to know which findings remain constant from system to system and which are unique to a particular CAI environment. Collection, retention and analysis of data on computer teaching/learning can increase our ability to study both the effectiveness of CAI and the details of the instructional process. Indeed the capacity for detailed study of the learning process is one of the more attractive features of CAI. Making comparisons based on the details of student response data from system to system will greatly accelerate our efforts to produced data-based, effective systems for automatic instruction. To accomplish this, however, we need to make an effort at standardizing our collection of data on student responses to instructional items.

Although our experience has been built on dealing with the detailed analysis of student responses to CAI learning situations in a number of curricular areas (elementary math drill and practice, arithmetic word problem solving, etc.), our current concern is with CAI in music. The development of CAI in music provides a particularly effective area in which to develop general data analysis and retention guidelines. Auditory perception of complex stimuli is a field in which much research remains to be done. Music CAI provides:

1. the technology for study of auditory perception;
2. a comprehensive set of stimuli in the curricula which can be adapted for research; and
3. a readily available pool of sophisticated subjects.

The comments we will make are, we feel, generally applicable but our specific examples will be built on our most recent experience through studies concerned with auditory perception in a CAI-Music environment.

As much, if not more, as with any area of research, replication is an important aspect of research in auditory perception. The study of music is not standardized on the pre-college level. Testing of auditory perception of musical concepts is not presently included in any of the current nationwide standardized tests. Facilities for such testing do not exist currently nor are they likely to be developed in the near future. Therefore replication of studies done at various locations will yield information for the establishment of base line data on auditory perception. Replication and continuation of studies will provide necessary longitudinal information.

II. DATA COLLECTION FOR REPLICATION.

Data retention and analysis must model both curriculum and subject to allow for replication. Although many, if not most, statistical treatments of the data will be planned in structuring these studies, data analysis may reveal configurations which suggest further modes of statistical analysis.

When data retention is modelled on the salient features of the stimuli presented, data can be re-formatted and rearranged so that further study of the results can be undertaken. This is especially important in cases where replications are structured to carry out an extension of previous studies. If the data retention is adequately modelled on the stimuli, then the researcher never faces the problem of being unable (due to lack of data retained) to investigate his data from a new viewpoint.

Data retention which models subject behavior aids substantially in thorough analysis. Research utilizing CAI is particularly capable of retaining information on subject responses. This allows study of interactive latency time, as well as study of the time necessary to complete a given study. Retention of all incorrect responses and student comments permits extensive analysis of subjects attitude and competence.

Taken as a total, data retention modelled on stimuli and subject response, rather than data retention sufficient only for the analysis of pre-planned statistical significance, provides greater flexibility within research objectives and establishes a basis on which to replicate and extend individual research efforts.

III. EXAMPLES OF REPLICATION.

At the Institute for Mathematical Studies in the Social Sciences (IMSSS), Stanford University, detailed data on student responses has been recorded and saved in machine-readable form (magnetic tape) for a number of years. What this has meant is that analysis procedures could be developed which require only minor modification as changing curricula were studied. In essence, the data saving has involved recording the stimulus, the student response, and the latency of the response for each item presented to a student. This constitutes a very large volume of data. Ordinarily, research is accomplished by collecting the data needed for the statistical design of a particular study. By recording the details of each student-curriculum interaction research questions formed well after the original study could be answered from the original raw data.

The programs for data retention and analysis for the CAI-Music project were adapted from this set of procedures and, as subsequent hypotheses were developed and tested, were further refined. The core of these analytic procedures was designed to take raw student response data and format it for analysis by relatively standard statistical routines. All curricula retained the same data in general - the stimulus presented and the student response. While specifics varied appropriate with each CAI course, conceptually the data generated is fairly consistent. The task is deciding what to view in any one analytic effort. Because the data is complete and constant, that task can be performed on any data at any time.

Item #	Order#	Stimulus Notes		mode	Response	Correct Answer	Eval	Time
1.	23	50	47	D	MI2	MI2	C	16:21:41
2.	13	51	47	D	MA2	MA2	C	16:20:23
3.	2	52	47	D	MI3	MI3	C	16:18:42
4.	25	53	47	D	MA3	MA3	C	16:21:50
5.	24	54	47	D	PE4	PE4	C	16:21:46
6.	40	55	47	D	MI6	T	X	16:24:06
7.	21	56	47	D	PE5	PE5	C	16:21:28
8.	48	57	47	D	MI6	MI6	C	16:25:28
9.	45	60	47	D	MI6	MA6	X	16:24:59
10.	37	61	47	D	T	MI7	X	16:23:41
11.	12	62	47	D	MA7	MA7	C	16:20:14
12.	28	63	47	D	MI6	PE8	X	16:22:02
13.	3	63	62	D	PE4	MI2	X	16:18:48
14.	41	63	61	D	MA2	MA2	C	16:24:30
15.	46	63	60	D	MI3	MI3	C	16:25:15
16.	19	63	57	D	MA3	MA3	C	16:21:18
17.	6	63	56	D	PE4	PE4	C	16:19:29
18.	42	63	55	D	T	T	C	16:24:33
19.	22	63	54	D	PE5	PE5	C	16:21:36
20.	31	63	53	D	MI6	MI6	C	16:22:56
21.	47	63	52	D	MA6	MA6	C	16:25:19
22.	5	63	51	D	MI7	MI7	C	16:19:05
23.	9	63	50	D	MI7	MA7	X	16:19:46
24.	34	63	47	D	MA7	PE8	X	16:23:26
25.	27	55	54	D	MI2	MI2	C	16:21:59
26.	26	55	53	D	MA2	MA2	C	16:21:54
27.	11	55	52	D	MI3	MI3	C	16:20:11
28.	18	55	51	D	MA3	MA3	C	16:21:14
29.	36	55	50	D	PE4	PE4	C	16:23:36
30.	43	55	47	D	T	T	C	16:24:43
31.	16	55	46	D	PE5	PE5	C	16:21:07
32.	29	55	45	D	PE5	MI6	X	16:22:26
33.	20	55	44	D	MA6	MA6	C	16:21:23
34.	10	55	43	D	MA7	MI7	X	16:20:05
35.	44	55	42	D	MA7	MA7	C	16:24:52
36.	33	55	41	D	PE8	PE8	C	16:23:20
37.	38	56	55	D	MI2	MI2	C	16:23:49
38.	35	57	55	D	MA2	MA2	C	16:23:33
39.	17	60	55	D	MI3	MI3	C	16:21:11
40.	32	61	55	D	MA3	MA3	C	16:23:16
41.	7	62	55	D	T	PE4	X	16:19:34
42.	4	63	55	D	T	T	C	16:18:53
43.	14	64	55	D	PE5	PE5	C	16:20:28
44.	8	65	55	D	MI6	MI6	C	16:19:40
45.	1	66	55	D	PE8	MA6	X	16:18:15
46.	39	67	55	D	MI7	MI7	C	16:23:53
47.	15	70	55	D	PE8	MA7	X	16:20:34
48.	30	71	55	D	PE8	PE8	C	16:22:41

Figure 1.
Data from one student on one topic in the CAI-Music Interval Study (Killam, Lorton & Schubert, 1975).

Data saved on the CAI-Music curriculum included the stimuli presented (pitches, speed of presentation, mode of presentation, etc.) and the student response (including the time of the response). Information on student input other than responses to stimuli (e.g., requests for new topics, repetitions of the stimuli, etc.) were also saved.

A sample of data retained, formated for ease in reading, for the interval study (discussed below) is in Figure 1.

Comparisons of four studies in musical perception will serve to illustrate ways in which data retention and analysis can assist a continuing research effort.

Plomp, et al. (1972), reported on a study of conservatory-trained S's identification of simple harmonic intervals presented at four different time durations: 15, 30, 60, and 120 msec. Data were presented in the form of a confusion matrix reproduced, in modified form, in Table 1.

The Stanford CAI-Music system was used to replicate and extend this study. The simple intervals were presented simultaneously and sequentially at 100 and 200 msec/pitch (Killam, Lorton & Schubert, 1975). To facilitate comparison with Plomp's study the Plomp, et al., data for 60 and 120 msec were extracted into separate confusion matrices and converted to percentages (see Table 2).

STIMULUS

	MI2		MA2		MI3		MA3		PE4		AU4		PE5		MI6		MA6		MI7		MA7		PE8	
MI2	33	25	7	8	0	2	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
MA2	6	6	35	28	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	1	0	0
MI3	0	0	1	2	40	34	2	7	1	1	3	0	0	0	0	0	0	0	0	1	0	0	0	0
MA3	0	1	1	1	3	4	41	33	1	0	0	0	1	0	1	1	3	1	0	0	0	0	0	0
PE4	0	0	0	2	0	2	1	3	41	39	4	5	6	7	1	2	0	0	4	0	1	1	0	0
AU4	0	3	0	1	0	1	0	0	1	1	35	35	0	1	0	1	0	2	0	3	0	0	0	0
PE5	0	0	0	1	0	1	0	1	1	3	0	0	34	35	0	0	0	0	0	0	0	0	1	0
MI6	0	0	0	0	0	0	0	1	0	1	1	1	1	2	39	32	3	3	3	0	0	0	0	0
MA6	0	0	0	0	1	0	1	0	0	0	0	1	1	0	2	6	38	36	2	2	0	1	0	0
MI7	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	1	1	0	31	31	2	3	2	1
MA7	6	7	1	1	0	1	0	0	0	0	1	2	2	0	0	2	0	3	2	6	41	33	1	4
PE8	0	3	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	6	41	40

Table 1.
Modified confusion matrix for 120 and 60 msec stimuli from Plomp, et al., 1973

		STIMULUS												
		MI2	MA2	MI3	MA3	PE4	T	PE5	MI6	MA6	MI7	MA7	PE8	TOT
	MI2	64	16	2							2			7
	MA2	13	70	1			1				1	2		7
	MI3		3	82	10	2	3				1			8
	MA3	1	2	7	82	1		1	2	4				8
R	PE4		2	2	4	88	10	14	3		4	2		11
E	AU4													
S	T	3	1	1		2	77	1	1	2	3			7
P	DI5													
O	PE5		1	1	1	4		76					1	7
N	MI6				1	1	2	3	78	6	3			8
S	MA6			1	1		1	1	8	82	4	1		8
E	MI7		1				1		2	1	68	5	3	6
	MA7	14	2	1			3	2	2	3	8	82	5	10
	PE8	3							1		2	6	90	8
	TOT	100	100	100	100	100	100	100	100	100	100	100	100	100

Table 2.
Plomp Confusion Matrix converted to percentages

By taking the relatively raw data from the Plomp, et al., report, making it machine readable, and putting it through the same analysis procedures as the data from the CAI-Music system rather direct comparisons could be made. Table 3 is the confusion matrix for the CAI-Music interval data. Figure 2 show one of the many direct comparative presentations which were possible by having part of the Plomp, et al., data on the IMSSS computing system.

A subsequent study of triad identification (Killam, 1976) was modelled on the interval study and run on the CAI-Music system at Stanford. Data retention allowed formatting in all modes of presentation, and also permitted extraction of triad data according to inversions of triads presented (see Tables 4 and 5).

		STIMULUS												
		MI2	MA2	MI3	MA3	PE4	T	PE5	MI6	MA6	MI7	MA7	PE8	TOT
	MI2	83	8									1		7
	MA2	12	80	2	1		1							8
	MI3	1	5	80	6	2	3	1	3	1				8
	MA3		1	7	84	5	2	3	1	2				9
R	PE4		1	2	1	82	6	8	11	1	1		1	9
E	AU4													
S	T						60		3		7	5		6
P	DI5						12							1
O	PE5			2		6	2	82	5	2	1		2	8
N	MI6				1		3		55	3	6	3	1	6
S	MA6			1	2				9	82	7	3	1	9
E	MI7						1		4	2	58	8	1	6
	MA7						3		2	1	11	70	2	7
	PE8							1	1		1	3	88	8
	TOT	100	100	100	100	100	100	100	100	100	100	100	100	100

Table 3.
Confusion Matrix from CAI-Music Study
(Killam, Lorton & Schubert, 1975)

RESPONSE	STIMULUS Diminished	 Minor	 Major	 Augmented	Percent of Total Response
Diminished	50	9	2	24	21
Minor	26	71	7	21	31
Major	11	12	88	10	30
Augmented	11	5	1	43	15
Total	98	98	98	97	97

Table 4.
Confusion Matrix of Triad Data, converted to percentages. (Killam, 1976)

Concurrently, Hofstetter (1975) independently developed the GUIDO system for CAI in music. He reported on triad identification in harmonic progressions, formatting the data on inversions and harmonic function identification in a confusion matrix.

Although the Stanford study of triad identification reports on triads accessed randomly and not in harmonic sequence, the data format of the Hofstetter and Stanford studies allows for potential comparison of inversion identification.

	Stimulus Root Position			First Inversion			Second Inversion			Augmented
	*DI	MI	MA	DI	MI	MA	DI	MI	MA	
RESPONSE	%	%	%	%	%	%	%	%	%	%
DI	58	8	1	44	8	3	47	12	2	24
MI	25	75	4	26	65	10	26	73	7	21
MA	9	10	92	12	17	83	12	10	87	10
AU	5	5	1	15	9	2	13	3	2	43

*DI = diminished, MI = minor, MA = major

Table 5.
Confusion Matrix of Triad Inversions (Killam, 1976)

```
                    COMPARATIVE HISTOGRAM

     .         20        40        60        80       100
MI2  .@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
     .****************************************

MA2  .@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
     .***************************************

MI3  .@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
     .***************************************

MA3  .@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
     .*****************************************

PE4  .@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
     .****************************************

T    .@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
     .***********************************

PE5  .@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
     .****************************************

MI6  .@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
     .**************************

MA6  .@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
     .****************************************

MI7  .@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
     .****************************

MA7  .@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
     .**********************************

PE8  .@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
     .*******************************************
```

Figure 2.
Comparative Histogram for Percent correct on each interval:
Plomp, et al (=p) and Killam, Lorton & Schubert (=*).

IV. DISCUSSION.

These four studies all consider auditory perception of Musical stimuli in a fairly standard setting. We have been able to make some very interesting comparisons from the confusion matrices in each study. However, because the original detailed stimulus-response data on the Stanford interval and triad studies were available for re-analysis, we were able to return to that data and accomplish a direct comparison of interval and triad perception (Lorton & Killam, 1976). This was useful and produced some interesting results.

A more important step would be to take, for example, the student response data from the Hofstetter study and extend our comparisons to results from a totally different computer system, musical tone production method and student population. To do this, we would need data recorded in sufficient detail to allow a complete regeneration of the instructional sequence. We would need the stimulus, the student response, an evaluation of that response and the time of the response. (We understand that such data does exist and are in the process of acquiring it.)

Most CAI systems have the capability to do precisely this task. PLATO, for example can save response data through the "output" and "dataon" TUTOR function. Even on small systems it is possible. Hewlett-Packard's Instructional Dialogue Facility (IDF) for the HP2000 Series Timesharing system has functions and procedures for saving student response data. These systems provide built in features for saving response data. However, any system can be adapted to do the same task. A CAI environment where student history is updated and retained can also save response data.

Response data, especially for an active CAI installation, can be massive. Often a judgement is made, by default if not explicitly, not to retain such data because no one has a use or need for it. It is very much needed if CAI is to evolve from the appealing toy stage and into the instructionally effective system development stage.

Obviously, the four studies cited above have much in common and comparative studies of them all yield important information. We are looking forward to opportunities for even broader comparisons between systems and locations.

The extensive development of Music CAI on the Plato system, especially by Hofstetter and his co-workers at the University of Delaware in their transfer of the GUIDO package to the PLATO system, offers an even greater opportunity for furthering this basic research in auditory perception in an instructional setting. As was mentioned above, the PLATO System has functions built in to retain the necessary data. We are pointing toward replications and parallel studies on the Stanford and Delaware systems and must find interesting similarities and differences. The key to the success of these studies is the consistent manner in which the response data will be retained.

V. CONCLUSION

We return to our opening question: "How do we know when CAI is effective?" The reason for such a question is that one goal in research on CAI is to produce effective systems of instruction. The systems developed from this work will be effective if they are built on reproducable results derived from studies which are not installation specific. One way to accelerate that effort is the ability to review student response data from a variety of systems so that hypotheses can be tested and verified across systems.

We feel our examples demonstrate the form of study which could be accomplished and indicate the type of data which should be retained. The refining of our understanding of the instructional process will occur as we are able to view and review student responses to elements in the computerized instructional sequence.

In summary then, our view is that Computer-Assisted Instruction programs destined to contribute to building toward a well tested and data based system for effective instruction must record data in sufficient detail so that the activity at the point of student-system interaction can be fully reproduced. This means four elements must be available in complete detail:

1. the stimulus presented
2. the student response
3. the evaluation of that response
4. the time of the response

With this information recorded and available for subsequent research and researchers, the refining of CAI into a system which effectively produces teaching/learning can be accomplished in a constantly advancing fashion.

REFERENCES.

Hofstetter, Fred J. GUIDO: An interactive computer-based system for improvement of instruction and research in ear-training. Journal of Computer Based Instruction, 1975, 1,100-106

Killam, R. N. A study of triad recognition and confusion by Stanford undergraduate music students. Unpublished DMA Final Project, Stanford University, 1976.

Killam, R. N., Lorton, P. & Schubert, E. Interval recognition: Identification of harmonic and melodic intervals. Journal of Musical Theory, 1975, 19, 212,234

Lorton, Paul & Killam, R. N. Interval and triad recognition: A comparison of two studies of musical perception. Paper presented to Second Annual Meeting of the National Consortium for Computer-Based Musical Instruction. Minneapolis, Aug. 1976

Plomp, R., Wagenaar, W.A. & Mimpen, A.M. Musical interval recognition with simultaneous tones. Acustica, 1973, 29, 101-109.

Bootstrapping CAI[1]

Richard G. Montanelli, Jr.
Department of Computer Science
University of Illinois at Urbana-Champaign

1. Introduction

CAI materials are rapidly being developed by enthusiastic proponents. Students are almost invariably very happy with the materials (often the materials are supplementary, and the students are using CAI for the first time) and report that their learning was enhanced. Then the proud developer marches off to a conference (ADCIS, AERA, ACM, AEDS, it doesn't matter which) and announces his or her findings to the world. Unfortunately the amount and type of data collected and published is usually not sufficient to support the claim that the materials teach, and thus should not convince prospective users of the materials that they are good.

2. Theoretical considerations

One logical way to prove or at least illustrate the effectiveness of instructional materials in a particular situation is to conduct a controlled experiment in which one-half of a group of prospective students is randomly assigned to study from the new materials while the other half uses the traditional materials. Then hopefully, comparisons between the two groups can provide data on the effectiveness of the new materials as compared with ordinary instruction. Unfortunately there are many drawbacks to this method: 1) It is expensive. Designing the experiment, obtaining instructor and student consent and support, randomly assigning students to sections, possibly adding or rescheduling sections, controlling other variables (instructor and meeting time, for example), monitoring the experiment, collecting data, and analyzing results are all time consuming and therefore expensive; 2) Students may become unhappy. In particular, if one group is perceived as getting better instruction, having more fun, or being subjected to machines, students may demand reassignment. (This author once spent over an hour with a student's father, a lawyer who maintained that his daughter's rights were being violated. Needless to say, the student was reassigned and dropped from the experiment.); 3) Results may not be very general, anyway. Results always refer only to the population of students sampled, but other lack of generality may exist also. Comparisons are ordinarily made against a particular instructor using particular materials, the instructor's own style, etc; and thus the "control group" may be highly unique. Also, perhaps the performance measures would not be universally accepted; 4) Controls are hard to maintain. Students will often find ways to study the material of the other group. If all else fails, they can look over the

---

[1] This research was supported in part by NSF Grant Nos. EC-41511 and EPP-74-21590.

shoulders of other students at the terminals. And how does one keep the "experimental" group out of the lectures?

On the other hand, one might argue that there is no need for any evaluation. Perhaps an analogy to textbooks might be drawn. Rarely are experiments conducted (or at least rarely published) to compare one textbook to another. Instructors may rely on their own evaluation of a new text or may also use evaluations and published reviews by colleagues. Could not this system work for CAI also? This author would maintain not, for several reasons: 1) It's easy to send an instructor a free or examination copy of a book, and he/she can take it home and examine it. Can the same be done with a computer terminal? 2) Teachers have had vast amounts of experience with books. They have been reading them all their lives. It is logical to conclude that as a result of this varied experience teachers can make reasonable evaluations and judgements as to the usefullness of particular texts for their students. However few if any teachers have equivalent experience with CAI, and not enough is yet known about what makes effective CAI. 3) A teacher may even try a new book, realizing that there are some risks. If the book does not depart too radically from the previous text, the teacher may invest little time and money (although the students probably have to buy the book). With CAI on the other hand, a large investment in hardware may be necessary. In any case, the instructors will have to undertake total reorganizations of their courses to take advantage of CAI, unless it is used solely as a supplement. Thus it is easy to see why instructors are often reluctant to plunge into CAI without some data on its effectiveness.

The purpose of this paper is to present and illustrate a strategy for documenting the effectiveness of CAI without going to the trouble of employing controlled experiments. Of course the data gathered may not be as conclusive or convincing to some as the results from a carefully designed and controlled experiment. However, results would certainly provide more data than is often available now. Additionally, the strategy provides quite a bit of data useful for lesson improvement.

The phrase "bootstrapping CAI" arose because the strategy was orginally designed for the dual purposes of evaluating CAI lessons and also for informing students as to their level of understanding of the CAI material. The strategy is as follows. First, lesson objectives are drawn up. These may be highly detailed, but this is not necessary. Then, the lesson and a quiz to test lesson objectives are implemented on the CAI system, by separate authors with instructions not to look at each other's work. (In the work done so far, most quizzes were actually written after the lessons had been used in classes. The only disadvantage was that some lesson improvements might have been made sooner if the quizzes had been programmed simultaneously with the lessons). Then both the lesson and the quiz are debugged, critiqued by faculty, revised, tried out by a few volunteer students, and generally evaluated in as many ways as possible to prepare for actual student use.

Finally, when both lesson and quiz are deemed ready, a class of students is selected to try them out. Of course, many types of problems with CAI materials can be corrected by having lesson authors and subject matter experts observe students using the materials.

Additional information can be gathered by collecting online data from student interaction with the materials. Barber (1975) gives a good example of this. There is no intent here to claim that the "bootstrapping" approach should replace these or any other valid lesson evaluation and improvement techniques. In fact quiz data may be most useful after a lesson has already undergone extensive improvements as a result of other forms of evaluation. In any case, eventually students use the lesson and take the quiz. Student answers are stored by question so that common errors can be found. Analysis of student errors in the quiz indicates areas of students' misunderstanding either of the quiz or of the lesson material. If, on restudy in light of analysis of student errors, a quiz question is found to be a good one, the presence of many student errors indicates problems in the lesson. Then the lesson can be revised and improved to provide better coverage of areas found inadequate by analysis of quiz data. Thus quiz data leads to lesson improvement which may also lead to quiz improvement, if ambiguous or poorly designed questions are uncovered (For example, if students continue to miss a question after much effort has been spent on improving the related area of the corresponding lesson.), and hence the term bootstrapping. A basic assumption of this technique is that it is easier for a potential user to evaluate a short quiz to see if a lesson covers the appropriate material than to examine the entire CAI lesson. A related assumption is that it is easier to judge "how well a lesson teaches" by looking at student responses to a quiz than by examining the lesson itself. Both of these assumptions were based mainly on this author's experience (Montanelli 1976 and 1977) with developing CAI materials over the last four years to teach introductory computer science.

3. Practical experience

In order to test out these ideas, a system for delivering quizzes and collecting data was designed for and implemented on Plato by Anderson (1976) in the fall of 1975. He also wrote the initial quiz (on FORTRAN character handling) which was used by about 25 students in a course at the end of that semester. Analysis of quiz data indicated lack of knowledge by students of the default length for CHARACTER variables, of the possibility of using other variable types for storing characters, and of the possibility of using special characters. (Note, students were learning a version of FORTRAN, WATFIV which includes CHARACTER declarations.) Although not fatal flaws (students could always explicitly declare variable lengths and use CHARACTER variables in their programs, probably desirable habits anyway), it was felt that students should know these other facets of character handling. So, the Plato lesson on FORTRAN/character handling (which had been previously used for several hundred students in introductory programming courses) was reexamined, and it was discovered that possibly in an effort to stress the sound programming practice of explicitly declaring the type for each variable, the lesson author had failed to mention the default length. Even worse, at a place where students were expected to type in a declaration statement in response to a question, the lesson judged a correct response using default typing as incorrect! Similarly, the lesson omitted any reference to use of other variable types and had few examples using special characters. Thus, extensive modifications were made to the lesson as a direct result of analysis of data collected from the quiz.

In the spring of 1976, several more quizzes were begun, although only one (on FORTRAN DO loops) was completed in time for classroom use that semester. In this instance, the DO loops lesson was relatively new, and had not been previously tested with a large number of students. Although several professors who taught the course involved had looked at the DO loop lesson and thought it acceptable (there were no factual errors; students could not get stuck or get an execution error), quiz results showed that several advanced concepts had been omitted from the lesson. In retrospect, the author (and the reviewers) had paid so much attention to the technique of presentation (animation), a not uncommon mistake when presenting material on a medium as powerful as graphics terminal, that not enough attention was paid to whether or not all required material was covered. In any case, students faired poorly on the quiz, and a major addition to the DO loop lesson was begun.

Currently about six quizzes are in use and all the FORTRAN lessons should have quizzes by the end of the school year.

4. Summary and conclusions

Results so far have shown that the bootstrapping approach described here is useful in improvement of CAI materials. Still to be tested is this author's belief that data obtained from student use of quizzes like those described here can be valuable as a part of the documentation and validation materials necessary for those considering using the CAI lessons. It seems unlikely that prospective CAI users would have the time, energy, or expertise to evaluate an entire CAI course. Then, in the absence of carefully controlled experiments, how can someone decide whether or not to use some CAI materials? The contention here is that when presented with a statement like, "When used by college freshmen in the College of Commerce and Business Administration at the University of Illinois, the following average quiz scores were obtained.", in the presence of the quizzes themselves and some summary statistics, prospective users of the CAI materials will be much more able to judge the effect of the materials on their students.

References

Anderson, R. I., "User's manual and guide to the ACSES quiz system," Report UIUCDCS-R-76-827, Department of Computer Science, University of Illinois at Urbana-Champaign, October 1976.

Barber, J. A., "Data collection as an improvement technique for PLATO lessons," Report UIUCDCS-R-75-777 (M.S. Thesis), Department of Computer Science, University of Illinois at Urbana-Champaign, December 1975.

Montanelli, R. G., Jr., "Evaluation of the use of CAI materials in an introductory computer science course," Presented at the AEDS International Convention, Phoenix, Arizona, May 1976.

Montanelli, R. G., Jr., "Using CAI to teach introductory computer programming," To appear in SIGCUE Bulletin, January 1977.